(19)日本国特許庁（ＪＰ）　(12)公表特許公報（Ａ）　(11)特許出願公表番号
特表2002－520748
（P2002－520748A）
(43)公表日　平成14年７月９日(2002.7.9)

(51)Int.Cl.'	識別記号	ＦＩ		テーマコード'(参考)
Ｇ０６Ｔ　15/00	２００	Ｇ０６Ｔ　15/00	２００	５Ｂ０５７
1/20		1/20	Ｃ	５Ｂ０８０

審査請求 未請求　予備審査請求 有　（全164頁）

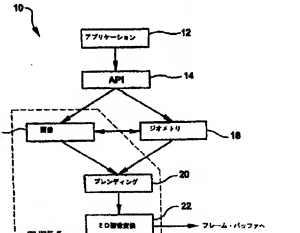
(21)出願番号　特願2000－560548(P2000－560548)
(86)（22)出願日　平成11年７月16日(1999.7.16)
(85)翻訳文提出日　平成13年１月12日(2001.1.12)
(86)国際出願番号　ＰＣＴ／ＵＳ９９／１６０３８
(87)国際公開番号　ＷＯ００／０４５０５
(87)国際公開日　平成12年１月27日(2000.1.27)
(31)優先権主張番号　６０／０９２，９７７
(32)優先日　平成10年７月16日(1998.7.16)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　ザ　リサーチ　ファウンデーション　オブ　ステイト　ユニヴァーシティ　オブ　ニューヨーク
アメリカ合衆国　ニューヨーク州　12201－0009　アルバニ　ピーオーボックス　９
(72)発明者　カウフマン　アリー　イー
アメリカ合衆国　ニューヨーク州　11803　プレインヴィュー　セダー　ドライヴ　ウェスト　94
(74)代理人　弁理士　中村　稔　（外９名）

最終頁に続く

(54)【発明の名称】　リアルタイム・ボリューム処理およびユニバーサル３Ｄレンダリングのための装置および方法

(57)【要約】
リアルタイム・ボリューム処理およびユニバーサル三次元レンダリング（１０）のための装置および方法である。この装置は、複数の三次元メモリ・ユニットと、大域水平方向連結を与える少なくとも１つのピクセル・パスと、複数のレンダリング・パイプラインと、少なくとも１つのジオメトリ・パスと、制御ユニットとを包含する。複数のレンダリング・パイプラインは、各々、好ましくは、補間、シェーディング、ＦＩＦＯバッファリング、連結およびルックアップ表のためのハードウェアを含む。本発明の装置は、表面、画像およびボリュームを単一の画像に混合するためのジオメトリ・パイプライン（１８）に接続してあってもよい。３Ｄボリュームのボリュメトリック・レイ・キャスティングを実施する方法は、所定の視点から主要投影軸線に沿って距離を計算する段階と、ボリュームを指数的に増大する境界を有する複数の連続領域に分割する段階と、ボリュームを通して視点から複数のレイをキャスティングする段階と、領域境界で２つまたはそれ以上のレイを合併するか、１つまたはそれ以上のレイを分割する段階と、ボリューム全体



【特許請求の範囲】

【請求項１】　リアルタイム・ボリューム処理およびユニバーサル三次元（３Ｄ）レンダリングのための装置であって、視点を定める視野・処理パラメータに応答するようになっており、また、前記視点から３Ｄボリューム投影画像または新しい修正ボリュームを生成し、この画像が複数のピクセルを含んでいる装置において、

複数の個別のボクセルを記憶する複数の三次元（３Ｄ）メモリ・ユニットであり、前記ボクセルの各々が、それと組み合わせた場所およびボクセル・データを有し、前記ボクセルが一緒になってボリューム・データセットを形成しており、前記視野・処理パラメータが、前記ボリューム・データセットの少なくとも１つのベースプレーンおよび前記ボリューム・データセットの最初と最後の処理スライスを定める三次元メモリ・ユニットと、

大域水平方向連絡を与える少なくとも第１のピクセル・バスと、

複数のレンダリング・パイプラインであって、各々が、前記複数の３Ｄメモリ・ユニットのうちの対応するものと前記少なくとも第１のピクセルの両方に垂直方向で接続されており、各々が、隣接のパイプラインとの水平方向連絡を有し、前記対応するメモリ・ユニットから前記ボクセル・データを受け取り、前記ボリューム・データセットの面と整合した二次元（２Ｄ）ベースプレーン画像を生成するレンダリング・パイプラインと、

前記複数のレンダリング・パイプラインとジオメトリ・エンジンとの間の大域水平方向連絡を与える少なくとも１つのジオメトリ・バスであり、単一画像においてジオメトリック、ボリュメトリックのオブジェクトを一緒にレンダリングするのを可能にする少なくとも１つのジオメトリ・バスと、

前記第１処理スライスを最初にサンプル・ポイントの現スライスとして指定し、前記最後の処理スライスに達するまで前記ボリューム・データセットの以降のスライスを通して現スライスとして走査することを制御する制御ユニットと、

を包含することを特徴とする装置。

【請求項２】　請求項１の装置において、さらに、前記複数の３Ｄメモリ・ユニットおよび前記複数のレンダリング・パイプラインに作動可能に接続した

フィードバック・バスを包含し、このフィードバック・バスが、前記レンダリング・パイプラインの出力部と、前記複数のレンダリング・パイプラインの入力部および前記複数の対応する３Ｄメモリ・ユニットのうちの少なくとも１つとの間の連絡を行うことを特徴とする装置。

【請求項３】　請求項２の装置において、さらに、前記複数の３Ｄメモリ・ユニットに作動可能に接続してあり、前記３Ｄメモリ・ユニット間の大域水平方向連絡を与える画像入力バスを包含することを特徴とする装置。

【請求項４】　請求項３の装置において、さらに、前記二次元（２Ｄ）ベースプレーン画像を記憶するための複数の二次元（２Ｄ）メモリ・ユニットを包含し、これら２Ｄメモリ・ユニットの各々が、前記複数のレンダリング・パイプラインの対応するものと前記少なくとも１つの第１ピクセル・バスの両方に接続していることを特徴とする装置。

【請求項５】　請求項４の装置において、さらに、前記少なくとも第１ピクセル・バスに作動可能に接続した少なくとも１つのワープ単位を包含し、この少なくとも１つのワープ単位が、所定の画像プレーン上へ前記ベースプレーン画像を組み立て、変換することを特徴とする装置。

【請求項６】　請求項１の装置において、前記複数のレンダリング・パイプラインの各々が、前記対応する１つに作動可能に接続した入力部と１つの出力部を有する少なくとも第１スライス・ユニットであり、前記３Ｄメモリ・ユニットからボクセル値を受け取る少なくとも第１スライス・ユニットと、

前記少なくとも第１スライス・ユニットの前記出力部に作動可能に接続した入力部と、前記少なくとも第１ピクセル・バスに作動可能に接続した１つの出力部とを有する合成ユニットと、

前記合成ユニットに作動可能に接続した入力部と１つの出力部とを有し、複数のピクセルを記憶し、これらのピクセルが、カラーおよびそれに組み合わせた不透明度を有する合成バッファと、

を包含することを特徴とする装置。

【請求項７】　請求項６の装置において、さらに、少なくとも第２、第３のスライス・ユニットを包含し、第１、第２、第３のスライス・ユニットが、前

記複数の３Ｄメモリ・ユニットから読み込んだ前記ボクセルを緩衝し、前記ボクセルの引き続く再読み込みを前記スライス・ユニットから実施することを特徴とする装置。

【請求項８】　請求項７の装置において、前記制御ユニットが、ブロック制御・介在要領で前記処理スライスの１横断分を組織化し、それによって、時間的に隣接したサンプル・ポイントが、主要投影軸線に沿った１つのユニット分異なることを特徴とする装置。

【請求項９】　請求項６の装置において、前記レンダリング・パイプラインの各々が、さらに、

前記合成バッファの前記出力部および前記合成ユニットに作動可能に接続したバイリニア補間ユニットと、

前記少なくとも第１スライス・ユニットの前記出力部に作動可能に接続した入力部と、１つの出力とを有するトリリニア補間ユニットと、

前記少なくとも第１スライス・ユニットの前記出力部に作動可能に接続した第１入力部および前記トリリニア補間ユニットの前記出力部に作動可能に接続した第２入力部を有し、そして、１つの出力部を有する勾配計算ユニットと、

この勾配計算ユニットの前記出力部に作動可能に接続した第１入力部および前記トリリニア補間ユニットの前記出力部に作動可能に接続した第２入力部を有し、また、前記合成ユニットの前記入力部に作動可能に接続した出力部を有するシェーディング・ユニットと

を包含することを特徴とする装置。

【請求項１０】　請求項９の装置において、前記複数のレンダリング・パイプラインの各々が、さらに、前記トリリニア補間ユニットおよび前記勾配計算ユニットを前記バイリニア補間ユニットに作動可能に接続するテクスチャ・マップ・バイパス・バスを包含することを特徴とする装置。

【請求項１１】　請求項９の装置において、前記複数のレンダリング・パイプラインの各々が、さらに、前記少なくとも１つのピクセル・バスと前記複数の３Ｄメモリ・ユニットの前記対応する１つと間の少なくとも１つのフィードバック・コネクションを包含し、前記フィードバック・コネクションが、前記少な

くとも１つのピクセル・バスから前記３Ｄメモリ・ユニットへのフィードバックを行い、次いで、前記複数のレンダリング・パイプラインにおける任意の中間ステージまでフィードバックを行うように構成してあることを特徴とする装置。

【請求項１２】　請求項９の装置において、前記勾配計算ユニットが、

乗算器と、

加算器とを包含し、

各主要投影軸線に沿って、３つの連続したサンプル・ポイントの加重平均を計算し、前記レンダリング・パイプラインおよび前記対応する３Ｄメモリ・ユニットに作動可能に接続したフィードバック・コネクションを通じて書き戻すことを特徴とする装置。

【請求項１３】　請求項１の装置において、前記制御ユニットが、前記ボリューム・データセットのビュー・フラスタムを、領域境界によって分離された複数の領域へ分割し、前記領域が、前記視点に最も近い第１領域を包含し、各連続した領域が、前記第１の領域で開始し、その前の隣接した領域のそれの２倍である、主要投影軸線に沿ったデプスを有することを特徴とする装置。

【請求項１４】　請求項１３の装置において、複数の連続したサイト・レイが、前記視点から前記ボリューム・データセットを通してキャストされ、前記複数のサイト・レイが、前記リューム・データセットを通過するときに互いに関して発散し、前記制御ユニットが、

(i) 前記サイト・レイのうちの少なくとも２つを結合すること、

(ii) 前記サイト・レイの少なくとも１つを分割すること

のうち一方を可能にすることを特徴とする装置。

【請求項１５】　三次元（３Ｄ）ボリューム・データセットの透視ボリュメトリック・レイ・キャスティングを実施する方法であって、前記ボリューム・データセットが、複数のボクセル・グリッド・ポイント場所を包含し、また、ビュー方向に対してほとんど直角な前記ボリューム・データセットの面を構成するベースプレーンを有する方法において、複数の三次元（３Ｄ）メモリ・ユニットに分散して記憶した個別のボクセルを利用し、前記ボクセルの各々が、それと組み合わせた場所およびボクセル・データを包含し、さらに、

（ａ）前記視点からの前記ボリューム・データセットの前記ベースプレーンまでの主要投影軸線に沿った距離を計算する段階と、

（ｂ）前記ボリューム・データセットを領域境界によって分離された複数の連続領域へ分割する段階であり、前記領域境界が、前記主要投影軸線に沿った前記視点からの指数的に増加する距離のところに位置しており、前記複数の領域が、少なくとも、ボリューム・データセットを含んでおり、前記第１領域後の各引き続く領域が、先の隣接した領域のデプスの２倍である、前記主要投影軸線に沿ったデプスを有し、

（ｃ）前記視点から前記ボリューム・データセットを通して延びる複数の連続したサイト・レイをキャスティングし、前記サイト・レイが、現サンプル領域を通過する段階と、

（ｄ）以下の１つを実施する段階であり、すなわち、

（ｉ）前記サイト・レイのうちの少なくとも２つを結合すること、および、

（ｉｉ）前記サイト・レイのうち少なくとも１つを分割することのうち１つを実施する段階と、

（ｅ）前記最終処理スライスに到達するまで前記ボリューム・データセットの、前記第１処理スライスで開始する引き続く領域を順次にサンプリングし、前記引き続く領域の各々が、順次に、現サンプル領域となるように段階（ｃ）、（ｄ）を繰り返す段階のうち１つを実施する段階とを包含することを特徴とする方法。

【請求項１６】　請求項１５の方法の透視レイ・キャスティング方法において、段階（ｃ）において、前記複数のサイト・レイが、前記ボリューム・データセットを通してキャストされ、前記サイト・レイの各々が、前記領域境界のところで前記ボリューム・データセットにおける前記ボクセル・グリッド・ポイント場所と一致することを特徴とする透視レイ・キャスティング方法。

【請求項１７】　請求項１５の透視レイ・キャスティング方法において、段階（ｄ）において、前記結合段階を、所定のフィルタ重みを有するフィルタで実施し、前記分割を、リニア補間で実施することを特徴とする透視レイ・キャスティング方法。

【請求項１８】　請求項１７の透視レイ・キャスティング方法において、

前記フィルタが、バートレット・フィルタであることを特徴とする透視レイ・キャスティング方法。

【請求項１９】　請求項１５の透視レイ・キャスティング方法において、段階（ｄ）において、複数の前記サイト・レイが、新しいサイト・レイを生成するように結合され、この新しいサイト・レイが、前記結合したサイト・レイの中心サイト・レイの正確な連続体であることを特徴とする透視レイ・キャスティング方法。

【請求項２０】　ソース・ボリューム・データセットのディテール・レベル（ＬＯＤ）表現を生成する方法であって、前記ボリューム・データセットが、複数の三次元（３Ｄ）メモリ・ユニットに分散して記憶された複数の個別のボクセルを包含し、前記ボクセルの各々が、それと組み合わせた場所およびボクセル・データを有する方法において、

（ａ）複数の新サンプルを生成するように前記ボリューム・データセットを再サンプリングする段階と、

（ｂ）前記複数の新サンプルのうち他のサンプルすべてを廃棄し、前記ソース・ボリューム・データセットよりサイズがコンパクトである第２ボリューム・データセットを形成する段階と、

（ｃ）前記第２ボリューム・データセットのスライスを前記３Ｄメモリ・ユニットに書き込み、それによって、前記ソース・ボリューム・データセットを前記第２ボリューム・データセットに置き換える段階と、

（ｄ）第２ボリューム・データセットが所定のサイズになるまで段階（ａ）～（ｃ）を繰り返す段階とを包含することを特徴とする方法。

【請求項２１】　請求項２０の方法において、段階（ａ）の再サンプリングをリニア補間を使用して実施することを特徴とする方法。

【請求項２２】　請求項２０の方法において、段階（ａ）の再サンプリングを、ｘ、ｙおよびｚ方向において０．５にセットした重みをもってトリリニア補間によって実施することを特徴とする方法。

【請求項２３】　ボリューム・データセットのディテール・レベル（ＬＯＤ）表現をレンダリングする方法であって、前記ボリューム・データセットが複

数の三次元（３Ｄ）メモリ・ユニットに分散して記憶された複数の個別のボクセルを包含し、各ボクセルが、前記ボリューム・データセット内のグリッド・ポイントに位置する場所を有し、また、それと組み合ったボクセル・データを有する方法において、

（ａ）視点および出発ディテール・レベル（ＬＯＤ）を定める視野・処理パラメータを選択する段階であり、前記出発ＬＯＤがスクリーン・スペース画像サイズにほぼ等しいベースプレーン画像を生成し、前記ベースプレーン画像が、視野方向に対して最も垂直である前記ボリューム・データセットの面を定めている段階と、

（ｂ）前記ボリューム・データセットを複数のスラブに分割し、各スラブが、前記視野方向に沿って、前記スラブの前後における投影ボクセル距離が２の因数だけ異なるようにする厚さを有し、前記複数のスラブが、前記ボリューム・データセットの最も詳細な表現の出発スラブを含んでいる段階と、

（ｃ）前記出発スラブを合成バッファ内にレンダリングし、この合成バッファが、前記描写スラブを表している合成バッファ画像を記憶する段階と、

（ｄ）前記合成バッファ画像を０．５の因数だけコンパクト化する段階と、

（ｅ）前記縮尺した合成バッファ画像を前記合成バッファへ書き込む段階と、

（ｆ）前記ボリューム・データセットのすべてのスラブが処理されてしまうまでボリューム・データセットの前記複数のスラブの各引き続くスラブを半解像度で順次に読み込むことによって段階（ｃ）、（ｄ）を繰り返す段階と

を包含することを特徴とする方法。

【請求項２４】　請求項２３の方法において、さらに、ワープ単位を用意する段階を包含し、段階（ｄ）を前記ワープ単位によって実施することを特徴とする方法。

【請求項２５】　複数のソース・ピクセルを包含しているソース画像および複数の目標ピクセルを包含している目標画像との間でジオメトリック変換を実施する順方向ワーピング方法であって、前記ソース画像を視点から生成する順方向ワーピング方法において、

前記視点および複数のスキャンラインを定める視野・処理パラメータを選ぶ段

階と、

平行・保存スキャンライン方向を決定する櫓家であり、前記ソース画像内の平行なスキャンラインが前記目標画像において平行のままとなるようにする段階と、

前記スキャンラインのすべてがマッピングされるまで前記ソース画像の引き続くスキャンラインを通して増分で走査することによって前記平行・保存スキャンライン方向に沿って前記目標画像に前記ソース画像をマッピングする段階と

を包含することを特徴とする順方向ワーピング方法。

【請求項２６】 請求項２５の順方向ワーピング方法において、前記平行・保存スキャンライン方向が、透視変換マトリックスに由来することを特徴とする順方向ワーピング方法。

【請求項２７】 請求項２５の順方向ワーピング方法において、さらに、

前記ソース画像を通して前記スキャンラインを走査する段階であり、各スキャンラインに沿ったサンプルを増分計算する段階と、

所定の読み込みテンプレートに従って固定パターンにおける前記ソース・ピクセルを読み込む段階と、

所定の書き込みテンプレートに従って前記ソース・ピクセルのサンプル値を書き込む段階と

を包含することを特徴とする順方向ワーピング方法。

【請求項２８】 請求項２７の順方向ワーピング方法において、さらに、前記ソース画像における隣接したサンプルまでの距離を算出することによってフットプリントを計算する段階を包含し、前記平行・保存スキャンライン方向に沿ったサンプルが等距離であり、各ソース・ピクセルの最も近い隣接目標ピクセルまでの距離を増分計算することを特徴とする順方向ワーピング方法。

【請求項２９】 少なくとも１つのボリュームおよび複数のジオメトリを含んでいるシーンをレンダリングし、画像を生成し、この画像を視点から生成する方法であって、

（ａ）複数のジオメトリ・プリミティブを用意し、前記ジオメトリ・プリミティブの各々がそれと組み合った不透明度を有し、また、複数のボクセルを含むボリ

ューム・データセットを用意し、各ボクセルが、それと組み合った場所とボクセル・データを有する段階と、

（ｂ）前記視点、前記ボリューム・データセットの少なくとも１つのベースプレーンであり、視野方向に対して最も垂直である前記ボリューム・データセットの面を定めるベースプレーンおよび前記ボリューム・データセットの最初と最後の処理スライスを定める視野・処理パラメータを選ぶ段階と、

（ｃ）前記不透明度に従って、半透明ジオメトリ・プリミティブから不透明ジオメトリ・プリミティブを分離する段階と、

（ｄ）前記不透明なジオメトリ・プリミティブをＺバッファへレンダリングする段階であり、前記Ｚバッファが、ボリューム・サンプル距離に一致するに充分な解像度を有し、前記視点から前記視野方向に沿った距離を表しているＺバッファ画像および前記ジオメトリ・プリミティブの各々についての関連したサンプル値を記憶し、前記Ｚバッファ画像が、前記ボリューム・データセットの前記第１処理スライスと一致するプレーン上にある段階と、

（ｅ）合成バッファを、前記不透明なジオメトリ・プリミティブの前記Ｚバッファ画像で初期化する段階と、

（ｆ）複数のサイト・レイを前記ボリューム・データセットの前記第１処理スライスを通してキャスティングする段階と、

（ｇ）視野方向に沿って採った、前記ボリューム・データセットの現サンプルのデプスが前記Ｚバッファ画像のデプスより大きいかどうかについて決定する段階と、

（ｈ）前記ボリューム・サンプルのデプスが前記不透明なジオメトリ・プリミティブのデプスより大きい場合に前記ボリューム・サンプル上に前記不透明なジオメトリ・プリミティブを合成する段階と、

（ｉ）前記最後の処理スライスに達するまで前記ボリューム・データセットの引き続くスライスを通して段階（ｆ）～（ｈ）を繰り返す段階と

を包含することを特徴とする方法。

【請求項３０】　請求項２９の方法において、さらに、前記半透明のジオメトリ・プリミティブを複数の薄いスラブにレンダリングし、各スラブが、すべ

てのスラブの合併が前記ボリューム・データセットの任意領域を失ったり、重複したりしないように選んだ境界を有するようにする段階と、

前記最後の処理スライスに達するまで前記ボリューム・データセットの２つの隣接したスライス間に前記半透明のジオメトリ・プリミティブの前記薄いスラブを合成する段階と

を包含することを特徴とする方法。

【請求項３１】　請求項３０の方法において、前記ボリューム・スライスおよび前記半透明のポリゴン・スラブからのデータを別の方法で相互にダブテールすることを特徴とする方法。

【請求項３２】請求項３０の方法において、半透明のジオメトリ・プリミティブをレンダリングする段階が、複数のバケットを創作し、各バケットが、前記ボリューム・データセットの２つの隣接したスライス間の薄いスラブに対応するようにする段階と、シーン内ですべての半透明のジオメトリ・プリミティブを移動させ、各半透明のジオメトリ・プリミティブを、前記半透明のジオメトリ・プリミティブが交差する各スラブのためのバケットに内に置く段階とを包含することを特徴とする方法。

【請求項３３】　請求項３０の方法において、半透明のジオメトリ・プリミティブをレンダリングする段階が、事前描写した半透明のジオメトリ・プリミティブの能動ディスプレイ・リストを維持する段階を包含し、それによって、半透明のジオメトリ・プリミティブの各々が、半透明のジオメトリ・プリミティブが交差する前記ボリューム・データセットの第１スライスで前記能動リスト内に置かれ、半透明のジオメトリ・プリミティブがもはやボリューム・データセットの任意のスライスと交差しなくなったときに前記能動リストから除かれることを特徴とする方法。

【請求項３４】　所定の視点に関してボリューム・データセットの任意の三次元回転を実施する方法であって、前記ボリューム・データセットが、複数の三次元メモリ・ユニット内に分散して記憶された複数の個別のボクセルを包含し、各ボクセルが、前記ボリューム・データセットのグリッド・ポイント上の場所を有し、それと組み合ったボクセル・データを有する方法において、

前記視点に関して前記ボリューム・データセットの最終的な向きを定める目標三次元回転マトリックスを受け取り、この回転マトリックスを、ｘ、ｙ、ｚ軸線に沿った軸線回転の連結として表す段階と、

前記回転マトリックスを前記ボリューム・データセットの連続的シャー変換のうち少なくとも２つのパスからなるシーケンスに分解する段階と

を包含することを特徴とする方法。

【請求項３５】 請求項３４の三次元回転方法において、分解段階が、以下のシーケンスでの連続的二次元（２Ｄ）シャー変換の４つのパスを包含し、このシーケンスが、

第１の主要方向軸線に沿ってボリューム・データセットの第１の２Ｄビーム・シャー変換を実施すること、

第２の主要方向軸線に沿ってボリューム・データセットの第２の２Ｄビーム・シャー変換を実施すること、

第３の主要方向軸線に沿ってボリューム・データセットの第３の２Ｄビーム・シャー変換を実施すること、そして、

前記第１の主要方向軸線に沿ってボリューム・データセットの第４の２Ｄビーム・シャー変換を実施すること

からなることを特徴とする三次元回転方法。

【請求項３６】 請求項３４の三次元回転方法において、分解段階が、以下のシーケンスでの連続的二次元（２Ｄ）シャー変換の４つのパスを包含し、このシーケンスが、

第１の主要方向軸線に沿ってボリューム・データセットの第１の２Ｄスライス・シャー変換を実施すること、

第２の主要方向軸線に沿ってボリューム・データセットの第２の２Ｄスライス・シャー変換を実施すること、

第３の主要方向軸線に沿ってボリューム・データセットの第３の２Ｄスライス・シャー変換を実施すること、そして、

前記第１の主要方向軸線に沿ってボリューム・データセットの第４の２Ｄスライス・シャー変換を実施すること

からなることを特徴とする三次元回転方法。

【請求項３７】 請求項３６の三次元回転方法において、分解段階が、連続的三次元（３Ｄ）ビーム・シャー変換の２つのパスを包含し、第１のパスが、前記第１、第２の２Ｄスライス・シャー変換の連結であり、第２のパスが、前記第３、第４の２Ｄスライス・シャー変換の連結であることを特徴とする三次元回転方法。

【請求項３８】 請求項３４の三次元回転方法において、分解段階が、以下のシーケンスでの連続的二次元（２Ｄ）シャー変換の３つのパスを包含し、このシーケンスが、

ボリューム・データセットの第１の２Ｄビーム・スライス・シャー変換を実施することを包含し、そこにおいて、前記シャー変換が、各々異なった腫瘍軸線方向に沿った、１つのビーム・シャー変換と１つのスライス・シャー変換の積であり、また、

ボリューム・データセットの第２の２Ｄビーム・スライス・シャー積変換を実施することを包含し、そこにおいて、前記シャー変換が、各々異なった主要軸線に沿った、１つのビーム・シャー変換と１つのスライス・シャー変換の積であり、さらにまた、

前記第１の２Ｄビーム・スライス・シャー変換のビーム・シャーと同じ主要軸線方向に沿ってボリューム・データセットの２Ｄビーム・シャー変換を実施すること

を包含することを特徴とする三次元回転方法。

【請求項３９】 請求項３４の三次元回転方法において、さらに、複数のバレル・シフタを提供する段階を包含し、前記分解段階の前記シャー変換が前記バレル・シフタによって実施されることを特徴とする三次元回転方法。

【請求項４０】 請求項３４の三次元回転方法において、さらに、複数の対数シフタを提供する段階を包含し、前記分解段階の前記シャー変換が前記対数シフタによって実施されることを特徴とする三次元回転方法。

【請求項４１】 請求項３４の三次元回転方法において、さらに、

（ａ）前記ソース・ボリューム・データセットのソース・ボクセルと目標ボリュ

ーム・データセットの目標ボクセルとの間に１対１の対応関係をセットアップする段階と、

（ｂ）前記ソース・ボクセルの各々について、前記１対１の対応関係を使用して前記対応する目標ボクセルを研鑽する段階と、

（ｃ）再サンプリングによってソース・ボリューム・データセットにおける前記ソース・ボクセル場所に前記目標ボクセルの各々を記憶する段階と、

（ｄ）前記目標ボクセルを前記目標ボリューム・データセットにおける対応する宛先へ変換する段階と

を包含することを特徴とする三次元回転方法。

【請求項４２】　請求項４１の三次元回転方法において、

段階（ｃ）における前記再サンプリングを、局所的ボクセルへの補間によって実施し、前記目標ボクセルの各々のサンプリング位置を、前記回転マトリックスの逆方向変換を使用して計算し、

段階（ｄ）における前記変換を最も近い隣接の実装機構を用いて実施することを特徴とする三次元回転方法。

【請求項４３】　複数のボリュームを含むシーンをレンダリングし、そして、１つの画像を視点から生成する方法において、

（ａ）第１のボリューム・データセットと、この第１のボリューム・データセットに対応する第１の変換マトリックスを用意する段階と、

（ｂ）第２のボリューム・データセットと、この第２ボリューム・データセットに対応する第２の変換マトリックスを用意する段階と、

（ｃ）前記視点および前記第１、第２のボリューム・データセットの最初と最後の処理スライスを定める視野・処理パラメータを選ぶ段階と、

（ｄ）それぞれ、前記第１、第２変換マトリックスに従って前記第１、第２ボリューム・データセットを変換する段階と、

（ｅ）前記第１のボリューム・データセットの前記最初の処理スライスを通して前記視点から視野方向に沿った複数のサイト・レイをキャスティングする段階と、

（ｆ）前記第２ボリューム・データセットの前記最初の処理スライスを通して前

記視点から前記視野方向に沿って複数のサイト・レイをキャスティングする段階と、

（ｇ）前記第１のボリュームの前記第１のスライスを合成バッファへ書き込む段階と、

（ｈ）前記第２ボリュームの前記最初のスライスを前記合成バッファ上に合成する段階と、

（ｉ）前記第１、第２のボリューム・データセットの前記最後の処理スライスに達するまで前記視野方向に沿って前記第１、第２のボリューム・データセットの引き続くスライスを通して順次に走査することによって段階（ｅ）～（ｈ）を繰り返す段階と

を包含することを特徴とする方法。

【請求項４４】請求項４３の方法において、段階（ｄ）が、さらに、前記第１、第２の変換マトリックス間の差を計算し、前記第１、第２のボリューム・データセットの一方のみを前記差に従って変換する段階を包含することを特徴とする方法。

【請求項４５】　請求項４４の方法において、段階（ｄ）が、さらに、シャー変換が前記ボリューム・データセットについて実施された場合に、前記ボリューム・データセットを再サンプリングして補正ボリューム・データセットを形成し、この補正ボリューム・データセットが、前記シャー変換により導入されたエラーをほぼ除去することを特徴とする方法。

【請求項４６】　請求項４５の方法において、前記再サンプリング段階を、段階（ｅ）、（ｆ）のうち少なくとも一方での前記レイ・キャスティング中に実施することを特徴とする方法。

【請求項４７】　複数のボリュームを含むシーンをレンダリングし、視点から１つの画像を生成する方法において、

（ａ）第１のボリューム・データセットおよび対応する第１の変換マトリックスおよび第２ボリューム・データセットおよび対応する第２の変換マトリックスを用意し、前記第１、第２のボリューム・データセットの各々が、複数の三次元メモリ・ユニットに分散して記憶された複数の個別のボクセルを包含しており、前

記ボクセルの各々が、前記ボリューム・データセットにおけるグリッド・ポイント上に位置する場所を有し、それと組み合わせたボクセル・データを有する段階と、

（ｂ）前記視点および前記第１、第２のボリューム・データセットの最初と最後の処理スライスを定める視点・処理パラメータを選択する段階と、

（ｃ）前記第１、第２のボリューム・データセットの前記最初の処理スライスを通して視野方向に沿って複数のサイト・レイをキャスティングする段階と、

（ｄ）前記サイト・レイに沿った各サンプル・ポイントでの前記第１、第２のボリューム・データセットからボクセルの対応する値を新しい値に結合することによって新しいボリューム・データセットを創り出し、前記新しい値の各々が、変調したカラー値を含む段階と、

（ｅ）段階（ｃ）を繰り返して、前記第１、第２のボリューム・データセットの前記最後の処理スライスにたっするまで前記ボリューム・データセットの引き続くスライスを通して走査することによって段階（ｃ）、（ｄ）を繰り返し、それによって、前記視点に対する単一のボリューム・データセットとして前記新しいボリューム・データセットをレンダリングする段階と

を包含することを特徴とする方法。

【請求項４８】　請求項４７の方法において、段階（ｄ）が、さらに、前記対応するボクセル値の各々の最大値を前記第１、第２のボリューム・データセットから計算する段階を包含し、ボクセルの結合を、前記最大値を、前記新しいボリューム・データセットにおける対応する新しい値として使用して合成することによって実施することを特徴とする方法。

【請求項４９】　請求項４７の方法において、段階（ｄ）が、さらに、前記第１、第２のボリューム・データセットからの前記対応するボクセル間の差値を計算する段階を包含し、ボクセルの結合を、前記新しいボリューム・データセットにおける対応する新しい値として前記差値を使用して合成することによって実施することを特徴とする方法。

【請求項５０】　請求項４７の方法において、段階（ｄ）が、さらに、前記第１、第２のボリューム・データセットからの前記対応するボクセルの各々の

最小値を計算する段階を包含し、ボクセルの結合を、前記新しいボリューム・データセットにおける対応する新しい値として前記最小値を使用して合成することによって実施することを特徴とする方法。

【請求項５１】　請求項４７の方法において、段階（ｄ）が、さらに、前記第１、第２のボリューム・データセットの各ボクセルに所定のＲＧＢα値を割り当てることによって前記第１、第２のボリューム・データセットを分類し、シェーディングする段階と、

対応するボクセルの前記ＲＧＢα値を変調してＲＧＢαボリューム・データセットを形成することによって前記第１、第２のボリューム・データセットの対応するボクセルを結合する段階と

を包含することを特徴とする方法。

【請求項５２】　請求項５１の方法において、段階（ｄ）が、さらに、前記新しいボリューム・データセットにおける前記ボクセルの各々に対応するインデックス・フィールドを創り出す段階を包含し、前記インデックス・フィールドが、前記新しいボリューム・データセットの発祥形態における各ボクセルをボリューム・データセット化することを示すデータを記憶することを特徴とする方法。

【請求項５３】　ｘ勾配成分、ｙ勾配成分およびｚ勾配成分のうち少なくとも１つを演算する段階を包含する、画像の高品質レンダリングを実施する方法において、

前記ｘ勾配成分が、（ｉ）重み－１、０、１をもってｘ方向に加重平均フィルタを適用するサブ段階と、（ｉｉ）重み１、ｗ、１をもってｙ方向に加重平均フィルタを適用するサブ段階と、（ｉｉｉ）重み１、ｗ、１をもってｚ方向に加重平均フィルタを適用するサブ段階とを包含し、

前記ｙ勾配成分が、（ｉ）重み－１、０、１をもってｘ方向に加重平均フィルタを適用する段階と、（ｉｉ）重み１、ｗ、１をもってｙ方向に加重平均フィルタを適用する段階と、（ｉｉｉ）重み１、ｗ、１をもってｚ方向に加重平均フィルタを適用する段階とを包含し、

前記ｚ勾配成分が、（ｉ）重み－１、０、１をもってｘ方向に加重平均フィル

タを適用する段階と、（ii）重み１、ｗ、１をもってｙ方向に加重平均フィルタを適用する段階と、（iii）重み１、ｗ、１をもってｚ方向に加重平均フィルタを適用する段階とを包含する

ことを特徴とする方法。

【請求項５４】　ジオメトリ・パイプラインに複数のＣｕｂｅ－５レンダリング・パイプラインを接続する方法であって、

スクリーン・リフレッシュ・ユニットにデータを供給するに充分なバンド幅を有する複数の外部記憶ユニットを設ける段階を包含し、

前記外部記憶ユニットがフレーム・バッファを形成し、このフレーム・バッファが、ボリューム・レンダリングおよびジオメトリ・レンダリングのうちの１つを実施するためのＲＧＢα、デプスおよびステンシル値のうちの少なくとも１つを記憶し、前記フレーム・バッファが、前記複数の記憶ユニットを横切ってインタリーブされており、Ｃｕｂｅ－５ボリューム・レンダリング・パイプラインに順番のアクセスで外部記憶ユニットのバンド幅の完全使用を可能にしており、さらに、

前記ジオメトリ・パイプラインが正しく前記フレーム・バッファにアクセスするように前記バッファ・インタリーブ構造の前記ジオメトリ・パイプラインに指示を与える段階を包含することを特徴とする方法。

【請求項５５】　ジオメトリ・パイプラインにＣｕｂｅ－５ボリューム・レンダリング・パイプラインを接続する方法であって、前記Ｃｕｂｅ－５レンダリング・パイプラインがフレーム・バッファを包含している方法において、

前記ジオメトリ・パイプラインから前記Ｃｕｂｅ－５パイプラインにジオメトリ・データを伝達する段階であり、前記ジオメトリ・データがラスタライズされたジオメトリの画像を含んでいる段階と、

前記Ｃｕｂｅ－５パイプラインにおける前記ジオメトリ・データを結合する段階と、

前記フレーム・バッファ内に前記結合されたジオメトリック・データを記憶する段階と

を包含することを特徴とする方法。

【請求項５６】　請求項５５の方法において、さらに、実行長コード化（ＲＬＥ）フォーマットで前記ジオメトリ・データをコード化する段階を包含することを特徴とする方法。

【請求項５７】　請求項５６の方法において、さらに、実行長コード化ラスタライゼーション・エンジンを設ける段階と、前記フレーム・バッファに前記ラスタライゼーション・エンジンを接続する段階とを包含することを特徴とする方法。

【発明の詳細な説明】

【０００１】

（政府権利声明）

本発明は、全米科学財団によって与えられた認可MIP9527694の下で、そして、海軍研究局によって与えられた認可N000149710402の下で政府支援により発明されたものである。政府は、この発明について一定の権利を有する。

【０００２】

（発明の背景）

本出願は、１９９８年７月１６日に出願された合衆国暫定特許出願番号第６０／０９２，９７７号に対する優先権を主張する。

【０００３】

（発明の分野）

本発明は、全般的に、三次元（３Ｄ）グラフィックスおよびボリューム視覚化に関する。一層詳しくは、本発明は、リアルタイム・ボリューム処理およびユニバーサル三次元レンダリングのための装置および方法に関する。

【０００４】

（従来技術の説明）

コンピュータ・レンダリングとは、情報の保全性および精度を維持しながら人間の思考にとって理解できるフォーマットに複雑な情報を変換するプロセスである。三次元現象に関する情報からなるボリュメトリック・データは、改良した画像レンダリング技術から利益を得ることができる複雑な情報の１種である。所与の視点からの、ボリュメトリック・データを提示するプロセスは、普通、ボリューム・レンダリングと呼ばれる。

【０００５】

ボリューム視覚化とは、スーパーコンピュータ・シミュレーションによって、あるいは、ボリューム・グラフィックス技術を使用してジオメトリ模型を合成することによって、入力装置（たとえば、生医学的スキャナ）で生成した大量のボリュメトリック・データを解釈する際に不可欠な技術である。ボリュメトリック・オブジェクトの操作、表示にとって特に重要なのは、映像及びレンダリング・

パラメータの対話的な変更、リアルタイムの表示速度、そして多くの場合、発展しつつある統合入力視覚化システムにおけるような、四次元（４Ｄ）視覚化（すなわち、時空的視覚化）と呼ばれる動的なデータセットの経時的変化を見ることが可能なことである。

【０００６】

ボリュメトリック・データセットは、普通、ボリューム要素（ボクセル）の３Ｄグリッドとして表され、しばしば、ボクセルのフル３Ｄラスタ（すなわちボリューム・バッファ）として記憶される。ボリューム・レンダリングは、ボリューム・データセットのグリッド・ポイントでボクセルによって表される連続的なオブジェクトや現象の３Ｄスカラー場を視覚化するための、最も普通の技術の一つであって、２つの主要な方法、すなわち、オブジェクト・オーダー（object-order）法および画像・オーダー（image order）法を使用して達成することができる。オブジェクト・オーダー法を使用すると、スクリーン・ピクセルに対する各ボクセルの寄与率が算出され、これらの寄与率を結合して最終的な画像が生成される。画像・オーダー法を使用すると、サイト・レイ（sight ray）が、ボリューム・データセットを通してスクリーン・ピクセルからキャスティングされ、これらのサイト・レイに沿ったボクセルの寄与率を用いて対応するピクセル値を評価する。

【０００７】

過去３０年にわたって、グラフィックス・システムは、倍々で発展してきた。初期の二次元（２Ｄ）から３Ｄ、４Ｄ（空間と時間）に発展し、ベクトル・グラフィックスからラスタ・グラフィックスへと発展し、ラスタ・グラフィックスでは、基本的グラフィックス・プリミティブとして、ベクトルがポリゴンに置き換えられた。このことは、毎秒数百万個の三角形を表示するのに適合化されたポリゴン・ベースのジオメトリエンジンの普及につながった。しかしながら、このようなシステムでは、三角形のファセットでオブジェクトの形状を近似させているだけである。３Ｄポリゴン・ベースのグラフィックス市場は、いまでにブームであり続け、パソコン（ＰＣ）業界の最も熱い競争の舞台の一つになっている。

【０００８】

伝統的なグラフィックス・システムに対する需要が急増しているのに応え、ジオメトリモデルの視覚的な現実感を向上させると共に、オブジェクト形状、構造を向上あるいは交換するために、不連続な画像を処理し表示するための種々の技術が考案されてきた。これら技術としては、２Ｄテクスチャ・フォト・マッピング、環境マッピング、画像ベース・レンダリング用のレンジ画像、２Ｄミップ・マッピング、ビデオストリーム、３Ｄボリューム、３Ｄミップ・マッピング、４Ｄ明視野・ルミグラフおよび五次元（５Ｄ）プレンオプチック（plenoptic）機能がある。これらすべての技術は、或る種の、不連続なピクセル、テクセル、ボクセルまたはｎオクセル間の次元ベース補間（バイリニア、トリリニア、クワッドリニア等）を必要とする。

【０００９】

ボリューム視覚化のための特殊な目的のコンピュータ・アーキテクチャおよび方法がこの技術分野では公知である。伝統的なボリューム視覚化方法は、代表的には、オブジェクトの正確な表現を提供するために逐次にボリューム・データセットを走査することによって作動する。たとえば、Dr. Arie Kaufman, Ingmar Bitter and Dr. Hanspeter Pfister（このうちの何人かは本願における発明者として名を連ねている）によって開発されたアーキテクチャ、Ｃｕｂｅ－４がある。Ｃｕｂｅ－４は、スライス・パラレル・レイキャスティングに基づく特殊目的スケーラブル・ボリューム・レンダリング・アーキテクチャである。Ｃｕｂｅ－４は、高解像度データセットの真のリアルタイム・レイキャスティング（たとえば、３０ヘルツのフレーム速度で1024^3の１６ビット・ボクセル）をデリバーすることができる。しかしながら、Ｃｕｂｅ－４は、透視投影についてはこのようなリアルタイム性能をデリバーすることはできない。現在のところ、公知の従来技術レンダリング・システムにおいては、透視投影を使用しても、レンダリング時間を増やすか、あるいは、投影画質を低下させるかのいずれかである。さらに、従来のアーキテクチャは、ボリュームおよびジオメトリを単一の画像に結合する能力を与えない。

【００１０】

次に図１を参照して、ここには、普通のボリューム視覚化システム１が示して

ある。図１に示すように、ボリューム・データは、ディスク２に記憶されており、レンダリング前にメモリ４にロードされる。次に、中央処理ユニット（ＣＰＵ）６が、メモリ４内に存在しているデータからボリューム描写画像を計算する。最終的な画像は、フレーム・バッファ８に書き込まれる。このフレーム・バッファは、代表的には、モニタ９または類似のディスプレイ装置上に表示するために、グラフィックス・カードに埋め込まれている。

【００１１】

したがって、本発明は、画像データ処理の新世代と考えられ得る程度まで、公知の方法および装置の能力をかなり向上させる方法および装置を提供することを目的とする。

【００１２】

他のおよび更なる目的は、本開示の結果として技術者の知るところであり、開示した発明の結果として実現されるすべての目的を含むことを意図している。

【００１３】

（発明の概要）

本発明は、網羅する新しい特性の故に、従来技術からの新発展に等しい。本発明によれば、リアルタイム・ボリューム処理およびユニバーサル三次元（３Ｄ）レンダリングのための装置は、１つまたはそれ以上の三次元（３Ｄ）メモリ・ユニットと、少なくとも１つの第１ピクセル・パスと、１つまたはそれ以上のレンダリング・パイプラインと、１つまたはそれ以上のジオメトリ・パスと、制御ユニットとを包含する。この装置は、視点を定める視野・処理パラメータに応答し、視点から３Ｄボリューム投影画像を生成する。投影画像は、複数のピクセルを包含する。

【００１４】

３Ｄメモリ・ユニットは、複数の個別のボクセルを記憶する。各ボクセルは、それと組み合わせた場所およびボクセル・データを有する。ボクセルは、一緒になって、ボリューム・データセットを形成する。そして、視野・処理パラメータは、ボリューム・データセットのベースプレーンとしてのボリューム・データセットの少なくとも１つの面と、ボリューム・データセットの最初と最後の処理ス

ライスとを定める。制御ユニットは、最初、サンプル・ポイントの現スライスとして最初の処理スライスを示し、最後の処理スライスに達するまで現スライスとしてボリューム・データセットの引き続くスライスを通しての走査を制御する。

【００１５】

レンダリング・パイプラインの各々は、垂直方向において、３Ｄメモリ・ユニットの対応する１つと少なくとも第１ピクセル・バスの両方に接続し、レンダリング・パイプラインの各々は、好ましくはせいぜい２つの最も近い隣接パイプラインとの大域的水平方向連絡を有する。レンダリング・パイプラインは、対応する３Ｄメモリ・ユニットからボクセル・データを受け取り、ボリューム・データセットの面と整合する二次元（２Ｄ）ベースプレーン画像を生成する。ジオメトリＩ／Ｏバスが、複数のレンダリング・パイプラインとジオメトリ・エンジンの間の大域的水平方向連絡を提供し、ジオメトリＩ／Ｏバスが、単一の画像におけるジオメトリック、ボリュメトリックのオブジェクトのレンダリングを可能にする。

【００１６】

本発明の装置および方法は、性能、画像レンダリング品質、融通性および単純性の向上のみならず、単一の画像におけるボリューム、表面（特に半透明性）を結合する能力に関しても、既存の３Ｄボリューム視覚化アーキテクチャおよび方法に勝る。本発明は、任意視野方向からの融通が利き、高品質な真のリアルタイム・レンダリングおよび高解像度データベースの内部、表面構造を視覚化する機構を提供する。そして、さらに、正確な透視投影、マルチ解像度ボリューム、多段オーバーラップ・ボリューム、クリッピング、勾配算出の向上、デプス・キューイング、ヘーズ、スーパーサンプリング、大きなボリュームの異方性データセットおよびレンダリングを含む様々なボリューム・レンダリング強化を支援する。

【００１７】

本発明は、単なるボリューム・レンダリング・マシンを超えたもの以上のものであり、高性能補間エンジンであり、たとえば、２Ｄ、３Ｄテクスチャ・マッピング（ミップ・マッピング）および画像ベース・レンダリングを含む補間にかな

り依存している個別の画像形成作業の高解像度ボリューム・レンダリングおよび加速のためのハードウェア・サポートを提供する。さらに、本発明の、ジオメトリ・エンジンと接続された装置および方法は、ボリュメトリック、ジオメトリックの方法を結合し、ユーザが、伝統的なジオメトリック・プリミティブ（たとえば、ポリゴンのファセット）、画像および単一の画像（ユニバーサル３Ｄレンダリングとして定義される）における画像、ボリュームを一緒に含む複雑なシーンを効果的に模型化し、レンダリングすることができる。

【００１８】

本発明の装置は、それに加えて、種々の大域的、局所的なフィードバック・コネクションを含むことによってシステム融通性を向上させ、これが、画像形成ワーピングおよび多解像度ボリューム処理のような進歩した画像形成作業を実施するようにパイプライン・ステージを再構成する能力を加える。さらに、本発明は、経済的な方法でこれらの目的を達成する。

【００１９】

本発明の利点は、ジオメトリ・エンジンに接続したとき、単一画像において一緒にジオメトリおよびボリュームの混合を可能にする方法および装置を提供することにある。

【００２０】

本発明の別の利点は、オーバーラップしているボリュメトリック・オブジェクトを効果的にレンダリングする方法および装置を提供することにある。

【００２１】

本発明のまた別の利点は、高度なボリューム・レンダリング機能を実施するために選択的なパイプライン・ステージを再構築する能力を有する装置を提供することにある。

【００２２】

本発明の更なる利点は、補間にかなり依存する強化像形成作業を支援する方法および装置を提供することにある。

【００２３】

本発明のまた更なる利点は、大きいボリューム・データセットの処理を加速す

る方法および装置を提供することにある。

【００２４】

本発明の別の利点は、既存の特殊目的ハードウェア・ボリューム視覚化システムよりも高い品質または迅速な透視投影を提供することにある。

【００２５】

本発明のさらに別の利点は、勾配評価、補間ユニットに対する強化と共に、画像の品質を容易に向上させることができる方法および装置を提供することにある。

【００２６】

本発明のまた別の利点は、ハードウェアで画像ワーピングを実施することのできる装置を提供することにある。

【００２７】

本発明のまた更なる利点は、公知のボリューム視覚化システムに固有の欠点を克服する方法および装置を提供することにある。

【００２８】

本発明のこれらおよび他の特徴および効果は、添付図面に関連して読むべき実施例の以下の詳細な説明から明らかになろう。

【００２９】

（好ましい実施例の詳細な説明）

本発明の装置および方法は、データを処理し、高解像度ボクセル・ベース・データセットのリアルタイム視覚化を支援することができる。本発明は、ジオメトリ（たとえば、ポリゴン）で画像（たとえば、ボリューム、テクスチャおよび画像）の統合に加えて強化されたボリューム・レンダリングをデリバーするユニバーサル三次元（３Ｄ）レンダリング・システムである。この装置および方法は、本願でも発明者として名を連ねているDr. Arie Kaufmanの発行済みの特許および係属中の出願、たとえば、米国特許第５，０３８，３０２号として１９９１年８月６日に発行された「Method of Converting Continuous Three-Dimensional Geometrical Representations Into Discrete Three-Dimensional Voxel-Based Representations Within a Three-Dimensional Voxel-Based System」、米国特

許第４，９８７，５５４号として１９９１年１月２２日に発行された「Method of Converting Continuous Three-Dimensional Geometrical Representations of Polygonal Objects Into Discrete Three-Dimensional Voxel-Based Representations Thereof Within a ThreeDimensional Voxel-Based System」、米国特許第４，９８５，８５６号として１９９１年１月１５日に発行された「Method and Apparatus for Storing, Accessing, and Processing Voxel-Based Data」、’５９３出願の継続出願として１９９３年３月１５日に出願された合衆国出願通し番号０８／０３１、５９９号によって放棄された合衆国出願通し番号０７／３４７、５９３号として１９８９年５月４日に出願された「Method of Converting Continuous Three-Dimensional Geometrical Representations of Quadratic Objects Into Discrete Three-Dimensional Voxel-Based Representations Thereof Within a Three-Dimensional Voxel-Based System」、米国特許第５，１０１，４７５号として１９９２年３月３１日に発行された「Method and Apparatus for Generating Arbitrary Projections of Three-Dimensional Voxel-Based Data」、合衆国出願通し番号０８／０９７、６３７号として１９９３年７月２６日に出願された「Method and Apparatus for Real-Time Volume Rendering From An Arbitrary Viewing Direction」、合衆国出願通し番号０７／８５５２２３番として１９９２年３月２０日に出願された「Method and Apparatus For Generating Realistic Images Using a Discrete Representation」および合衆国出願通し番号０８／９１０、５７５号として１９９７年８月１日に出願された「Apparatus and Method for Parallel and Perspective Real-Time Volume Visualization」に記載されているようなボクセル・ベース・システムとして使用できるように設計してある。これらの引用文献の全開示内容は、参考資料としてここに援用する。

【００３０】

図２は、本発明の一実施例に従って形成したユニバーサル３Ｄレンダリング・システム１０の概念図を示す。レンダリング可能なオブジェクトの集合体を表示するアプリケーション１２は、アプリケーション・プログラム・インタフェース（ＡＰＩ）１４によって適切な画像およびジオメトリ表現に分割されると好まし

い。これらの表現は、それぞれ、引き続いて、画像ユニット１６およびジオメトリ・ユニット１８によって処理される。これらのユニットは、機能ブロックとして全体的に図示してある。画像ユニット１６は、好ましくは、複数の画像パイプライン（図示せず）を包含し、ジオメトリ・ユニット１８は、好ましくは、それぞれ、画像およびジオメトリ表現をレンダリングするための複数のジオメトリ・パイプラインを包含する。画像ユニット１６およびジオメトリ・ユニット１８の描写出力は、ブレンディング・ユニット２０において結合され、単一のベースプレーン画像を生成する。このベースプレーン画像は、好ましくは、表示のためにワープ単位２２によって最終的な投影プレーンに変換されるとよい。

【００３１】

図３は、本発明のＣｕｂｅ－５ボリューム視覚化システムの１つの実施例を示す。図３に示すように、このシステムは、好ましくは、１つまたはそれ以上の三次元メモリ・ユニット２４を包含し、各３Ｄメモリ・ユニット２４が、入力バス２６および対応するＣｕｂｅ－５チップ２８に垂直方向に接続される。複数のＣｕｂｅ－５チップ２８が、フレーム・バッファ・ピクセル・バス３４に接続した状態で示してある。さらに、本発明のシステム１０は、好ましくは、少なくとも１つの普通のジオメトリ・エンジン３０とホストコンピュータ３２に接続しており、これらは、共に、本発明のＣｕｂｅ－５装置と連絡するために入力バス２６とフレーム・バッファ・ピクセル・バス３４との間に作動可能に接続されている。

【００３２】

次に図４を参照して、本発明の装置１０は、画像入力バス２６に好ましくは接続した複数の３Ｄメモリ・ユニット２４を包含しており、３Ｄメモリ・ユニット２４間の大域水平方向連絡を行うようになっている。ボリューム・データセットは、普通、フル３Ｄラスタ（すなわち、ボリューム・バッファ）としてしばしば記憶されるボリューム要素すなわちボクセルの正規グリッドとして表される。このボリューム・データセットは、好ましくは、３Ｄメモリ・ユニット２４を横切って分散されると好ましい。スキュード分布の場合、本発明は、主要軸線の内の任意の軸線に対して平行してボクセルの完全なビーム（すなわち、列）への競合

なしアクセスを可能にする。図４に示すように、システム１０を通してビデオまたは４次元（４Ｄ）ボリューム・データをストリーミングするために、各３Ｄメモリ・ユニット２４は、好ましくは、専用のリアルタイム入力部３６に接続される。リアルタイム入力ソースに専用の接続を行うことによって、メモリ・プロセッサ・バンド幅によるボトルネックがさらに軽減される。

【００３３】

本発明のユニバーサル３Ｄレンダリング・システム１０は、さらに、複数のレンダリング・パイプラインを包含する。これらは、図４においてＣｕｂｅ－５ユニット３８の機能ブロックとして示してある。各レンダリング・パイプライン３８は、対応する３Ｄメモリ・ユニット２４に接続してあり、好ましくは、少なくとも好ましくはその２つの最も近い隣接３Ｄメモリ・ユニットと水平方向に連絡する。Ｃｕｂｅ－５ユニット３８は、それらの専用の３Ｄメモリ・ユニット２４から読み出し、二次元（２Ｄ）ベースプレーン画像を生成する。このベースプレーン画像（Ｃｕｂｅ－５ユニット３８によって生成した複数の合成ピクセルを含む）は、複数の二次元（２Ｄ）メモリ・ユニット４０を横切って分散されると好ましい。２Ｄメモリ・ユニット４０の各々は、好ましくは、２Ｄメモリ・ユニット４０間の大域水平方向連絡を提供する対応するＣｕｂｅ－５パイプライン・ユニット３８およびベースプレーン・ピクセル・バス４２に接続している。

【００３４】

好ましくは、本発明は、ベースプレーン・ピクセル・バス４２に接続された複数のワープ単位４４を包含する。ワープ単位４４は、複数の２Ｄメモリ・ユニット４０に記憶されたベースプレーン画像を組み立て、ユーザ定義画像プレーン上へ変換する（すなわちワープする）。本発明は単一のワープ単位４４を使用すること（たとえば、ハードウェアのコストまたはオーバーヘッドを低減するために）を意図しているが、複数のワープ単位４４が画像変換を加速すると望ましい。

【００３５】

各ワープ単位４４の出力部は、好ましくは、フレーム・バッファ・ピクセル・バス３４に接続してあり、ワープ単位４４間の大域水平方向連絡を行う。ベースプレーン・ピクセル・バス４２を通してソース・ピクセルを読み込むことと、フ

レーム・バッファ・ピクセル・バス３４を通して最終的な画像ピクセルを書き込むことは、同時に生じ、より大きいシステム・スループットを可能にすると好ましい。好ましいアーキテクチャではないが、本発明は、また、ワープ単位４４による逐次的読み書きも意図している。こうすれば、この１つのピクセル・バスが、フル・スクリーン画像のためのリアルタイム画像転送にとって充分なバンド幅を提供すると仮定してのことだが、ピクセル・バスがたった１つでもよい。

【００３６】

図４を続けて参照して、本発明は、好ましくは、ジオメトリ入力バス４６とジオメトリ出力バス４８とを包含するが、２つのバスを、リアルタイム像形成にとって充分なバンド幅の単一のジオメトリ入出力バスにまとめることも考えられる。ジオメトリ入力、出力バス４６、４８は、それぞれ、Ｃｕｂｅ－５ユニット３８の入力部、出力部に接続してあり、少なくとも１つのジオメトリ・パイプラインまたはエンジン（図示せず）を本システム１０に独特な方法で接続すると好ましい。本発明のアーキテクチャ（ジオメトリ・バス４６、４８を経てジオメトリ・エンジンに接続している）は、画像（たとえば、ボリュームおよびテクスチャ）のジオメトリ（たとえば、ポリゴンおよび表面）との統合をサポートする。ジオメトリ・でえたのボリュウメトリック・オブジェクトとのこの混合は、本発明に特有の強力な特徴である。

【００３７】

次に図５を参照して、ここには、本発明の一実施例に従って形成された複数のＣｕｂｅ－５レンダリング・パイプライン（図４の参照コード３８）のうちの１つの機能ステージを表すブロック図が示してある。図５に示すように、各レンダリング・パイプライン５２は、好ましくは、４種類の処理ユニットを包含する。すなわち、トリリニア補間ユニット（TriLin）５４、勾配評価ユニット（Gradient）５６、シェーディング・ユニット（Shader）５８および合成ユニット（Compos）６０である。これらレンダリング・パイプライン・ステージの各々は、従来の立方体ボリューム視覚化アーキテクチャ（先に延べた）に関するArie Kaufmanの先に発行された特許および係属中の出願に詳しく記載されており、したがって、ここでは簡単に以下に説明する。

【００３８】

３Ｄメモリ・ユニット２４に関して上述したように、ボリューム・データセットは、対応する３Ｄメモリ・ユニット２４（図４参照）に接続した各Ｃｕｂｅ－５ユニット３８と共に、３Ｄメモリ・ユニット２４を横切ってスキューした状態で分散したボクセルの正規グリッドとして記憶される。同じスキューしたビームのボクセルは、好ましくは、並列に取り込まれて処理され、すべてのＣｕｂｅ－５ユニット３８を横切って分散させられる。予め定めたベースプレーンに対して平行な（すなわち、所定の視野方向に対して最も垂直であるボリューム・データセットの面と平行な）ボリューム・データセットの連続したスライスは、好ましくは、スキャンライン・オーダーにおいてトラバースされる。再び図５を参照して、アドレス生成・制御ユニット６２が、好ましくは、３Ｄメモリ・ユニット２４内へアクセスするためのアドレスを生成する。さらに、アドレス生成・制御ユニット６２は、現処理スライスとして最初の処理スライスを指示し、最終的なスライスが処理されてしまうまでボリューム・データセットの引き続くスライスを通しての走査を制御する。

【００３９】

トリリニア補間ユニット５４は、２つの処理スライス間の補間サンプル値の新しいスライスを計算する。あるいは、トリリニア補間機能を一連のリニアまたはバイリニア補間として実施することも考えられる。

【００４０】

勾配評価ユニット５６は、好ましくは、ボリューム・データセットの複数のスライスからのボリューム・データを使用して中央差分勾配を計算する。勾配評価ユニット５６によって生成した中央差分勾配を利用して、引き続き、現処理スライスのサンプル・ポイントにシェーディング・ユニット５８が陰影を付ける。シェーディング・ユニット５８は、好ましくは、１つまたはそれ以上のルックアップ表（ＬＵＴ）（好ましくは、材料カラーおよび強度情報を記憶している各シェーディング・ユニット５８に存在するもの）へのインデックスとしてサンプルおよび勾配を使用する。材料カラー表は、データセット・タイプ依存であり、カラー強度表は、当業者には公知のように、局所的照明模型に基づいている。簡単に

言えば、カラーおよび強度の乗算が、各サンプルについてのピクセル・カラーを生成し、これが、合成ユニット６０において使用され、このカラーを各サイト・レイに沿った先に蓄積されていたピクセルと合成する。

【００４１】

再び図４を参照して、連続するサイト・レイに沿った次のサンプルを計算するためのデータは、隣接したＣｕｂｅ－５ユニット３８上に存在する可能性がある。この場合、Ｃｕｂｅ－５ユニット３８間の最も近い隣接のコネクションが、好ましくは、必要なデータを適切なＣｕｂｅ－５ユニット３８に送るのに用いられ、このＣｕｂｅ－５ユニット３８が、この特定のサイト・レイを処理し続けることになる。合成が完了したとき、合成されたピクセル（すなわち、ベースプレーン・ピクセル）は、好ましくは、Ｃｕｂｅ－５ユニット・パイプライン３８に接続した対応する２Ｄメモリ・ユニット４０に記憶される。ベースプレーン・ピクセル（ベースプレーン画像を形成している）が、その後、ベースプレーン・ピクセル・バス４２を経て２Ｄメモリ・ユニット４０から読み込まれ、ワープ単位４４によって組み立てられる。ワープ単位４４は、それに加えて、最終的な投影プレーン画像にベースプレーン画像を変換する。

【００４２】

図５を参照して、トリリニア補間ユニット５４および勾配評価ユニット５６にとって必要なデータ遅延は、好ましくは、トリリニア補間５４および勾配評価ユニット５６によって処理される前に１つまたはそれ以上の先入れ先出し（ＦＩＦＯ）ユニット６４をパイプライン・データ経路に挿入することによって達成される。ＦＩＦＯユニット（単数または複数）６４は、たとえば、ランダム・アクセス・メモリＲＡＭ（好ましくは、Ｃｕｂｅ－５チップに埋め込まれている）として実装してもよい。ボリューム・データセットの複数のスライスのビームを同時に処理するとき、したがって、スライス間により長い計算時間が必要なときには、所定遅延の導入は特に重要である。

【００４３】

バイリニア補間ユニット（ＢｉＬｉｎ）７２に作動可能に接続した合成バッファ（Ｃｏｍｐｏｓ Ｂｕｆｆｅｒ）７４が、本質的に、１つのスライスＦＩＦＯを提供する。バイリ

ニア補間ユニット７２は、好ましくは、テクスチャ・マッピングのために必要に応じてボクセル間の値を得るように補間する。ボリューム・レンダリングのために、BiLin７２は、好ましくは、ボリューム・データセットのコーナー・ボクセルの１つを選ぶ（Select ｘおよびselect ｙによって決定される）０．０または１．０の重みだけを使用する。レイがパイプラインと交差する場合、レイ・データだけを移動させる。ｘおよびｙについてのｍｕｘ[?]だけで、ボリューム・レンダリングについては充分であるが、テクスチャ・マッピングにとってはバイリニア補間が好ましい。

【００４４】

Ｃｕｂｅ－５アーキテクチャは、好ましくは、パイプライン・ステージの再順序付けおよびマルチパス・レンダリング、処理動作の数をサポートする。この場合、Ｃｕｂｅ－５レンダリング・パイプライン５２および３Ｄメモリ・ユニット２４の種々のステージ間にフィードバック・コネクションが必要となる。たとえば、オーバーラップしているボリュメトリック・オブジェクトの正しいレンダリングには、好ましくは、Ｃｕｂｅ－５パイプライン５２を通る少なくとも２回のパスが必要であり、このとき、一回目のパスでボリュメトリック・オブジェクトを再サンプリングして互いに整合させ、二回目のパスでインタリーブ・オーダー式にボリュメトリック・オブジェクトをレンダリングする。図５に示すように、多ボリューム・フィードバック経路６６を設けると好ましく、これが合成ユニット６０の出力部を対応する３Ｄメモリ・ユニット２４に作動可能に接続する。これにより、再サンプリング・ボリュームを、再サンプリング、分類、シェーディング後に３Ｄメモリ・ユニット２４に書き戻すことができる。最終的なレンダリング・パスは、ＲＧＢαボリュームについて作動する。

【００４５】

同様に、各Ｃｕｂｅ－５レンダリング・パイプライン５２は、好ましくは、ワープ単位４４と３Ｄメモリ・ユニット２４の間に接続した画像ベースのレンダリング・フィードバック経路６８を包含する。画像ベースのレンダリング・フィードバック・ライン６８は、好ましくは、中間ワープド画像を３Ｄメモリ・ユニット２４に書き込むためのフィードバック経路を提供する。これは、特に、複数の

ワープ・パスを必要とする或る種の画像ベース・レンダリング作業を加速するために有用であるかも知れない。本発明のアーキテクチャは、さらに、３Ｄメモリ・ユニット２４と他の種々のＣｕｂｅ－５レンダリング・パイプライン・ステージとの間、または、個々のパイプライン・ステージそれ自体の間のフィードバック・コネクションも意図している。画像レンダリング速度は、フィードバック経路を含むことによってかなり速くなり、これは、Ｃｕｂｅ－５レンダリング・パイプライン５２全体を通して移動するように結果を待つことなく、個々のパイプライン・ステージの計算結果への直接的で即時のアクセスを提供する。

【００４６】

本発明の好ましい実施例においては、Ｃｕｂｅ－５システムは、レンダリング・パイプラインの選択的なステージを迂回するコネクションを包含する。これは、たとえば、或る種の像形成作業にとっては不要であるかも知れない。これらの未使用のパイプライン・ステージを迂回することによって、このような像形成作業が加速される可能性はある。図５に示すように、テクスチャ・マップ・バイパス７０は、好ましくは、各Ｃｕｂｅ－５レンダリング・パイプライン５２に含まれる。このテクスチャ・マップ・バイパス・コネクション７０は、実質的にミップ・マッピングの速度を上げる。このミップ・マッピングは、たとえば、シェーディング・ユニット５８および合成ユニット６０を迂回し、トリリニア補間ユニット５４および勾配評価ユニット５６からの結果をバイリニア補間ユニット７２に直接与えることによって処理されるべき画像の複数のディテール・レベル（ＬＯＤ）を記憶することからなる。このような方法において、本発明のアーキテクチャは、好ましくは、ボリューム・レンダリングを実施するためのパイプライン・アレイとしてばかりでなく、様々な像形成作業を実施するように選択的に構成し得るハードウェア・リソースの集まりとしても考えることができる。たとえば、本発明のＣｕｂｅ－５システムがボリューム・レンダリングを実施しているとき、本質的に、ハードウェア・リソースのすべてが必要であるのに対し、テクスチャ・マッピングは、一般的に、メモリ、或る種のバッファリングおよび補間ユニットだけしか必要としない。

【００４７】

以下に説明する本発明の別のユニークで重要な局面は、Ｃｕｂｅ－５アーキテクチャが少なくとも１つの普通のジオメトリ・エンジン７６とインタフェース接続していて、単一画像におけるジオメトリック・データとボリュメトリック・オブジェクトとの混合をサポートすることができるということにある。これは、好ましくは、上述したように、少なくとも１つのジオメトリ・バスを設けてジオメトリ・エンジン７６とインタフェース接続させることによって達成される。

【００４８】

好ましくは、本発明のＣｕｂｅ－５アーキテクチャは、可能なときにはいつでも、パイプライン構成要素（たとえば、補間ユニット等）を再利用し、複数の構成、特に、複数のボリュメトリック・オブジェクト、ポリゴン・オブジェクトのシーンのレンダリング、テクスチャ・マッピングおよび画像ベース・レンダリングを使用して様々なレンダリング・アルゴリズムを加速するようになっている。他の重要な利点としては、パイプライン構成要素を再利用することで、ハードウェア・コストを低減できるということがある。Ｃｕｂｅ－５アーキテクチャは、また、ボリューム・レンダリングを向上させ、他の像形成作業を加速させるための種々のユニークな方法およびアルゴリズムをサポートする。これらの方法およびアルゴリズムのうちのいくつかを、以下に個別にさらに詳しく説明する。

【００４９】

本発明に従って形成した、Ｃｕｂｅ－５システムの好ましい実施例においては、ボリューム・データセットはブロックとして記憶され、それによって、空間的場所についての利点を得ている。リニア・ブロッキング（たとえば、ボクセレータＡＰＩ）の代わりに、階層的なブロックを使用している。これは、好ましくは、複数の３Ｄメモリ・ユニットを横切って分散された配置で記憶され、スキューさせられる。たとえば、現三菱電機１６ビット、１２５メガヘルツ同期ダイナミック・ランダム・アクセス・メモリ（ＳＤＲＡＭ）を使用して３Ｄメモリを実装した場合、各ブロックは、１０２４バイトを必要とする$8^3$１６ビット・ボクセルまたは２つのＳＤＲＡＭページを収容し得る。

【００５０】

各ブロックは、好ましくは、同じ３Ｄメモリ・ユニットに存在している2^3ボ

クセル・ミニブロックとして構成される。ＳＤＲＡＭ内部のバンクは、好ましくは、パイプライン式にアクセスされ、現バースト転送がほぼ完全に以降のバースト転写のセットアップを隠すようにすることができる。ミニブロック内のボクセルのビュー依存処理オーダーがそれらの記憶オーダーと一致しない場合には、８つのミニブロック・ボクセルをＣｕｂｅ－５チップで再オーダーすると好ましい。それ故、ＳＤＲＡＭ上のボリューム・データセットのコピーは、たった１つで充分である。したがって、階層的なブロッキングにより、ほぼフル・バースト・モード速度、ほぼフル（１００％）バンド幅利用、ビュー独立データ記憶およびバランスの取れた作業負荷で、ミニブロックにランダム・アクセスすることが可能になる。

【００５１】

ブロッキングは、メモリ・インタフェースを最適化するだけではなく、チップ間連絡バンド幅（すなわち、Ｃｕｂｅ－５ハードウェア・ユニット間の）を縮小するという付加的な利点も有する。これは、ブロック周辺部にあるボクセルだけが隣接したチップス処理隣接ブロックとの間で交換される必要があるからである。Ｏ（b^3）時にb^3ボクセル・ブロックを処理しながら、ブロック境界上のＯ（b^2）ボクセルはチップス処理隣接ブロック間で連絡する必要があるだけである。ここで、ｂはブロック・エッジのサイズであり、各ブロックは、ｂ×ｂ×ｂ（すなわち、b^3）のボクセルを有する。したがって、チップ間連絡は、ノンブロッキング解の場合よりもＯ（１／ｂ）小さいバンド幅でよい。ブロック・エッジのサイズｂは、約４＝ｂ＝６４の範囲にあってもよいが、８のブロック・エッジ・サイズが好ましい。

【００５２】

ブロック・ルックアップ表（ＬＵＴ）は、現ボリュームを含むすべてのブロックへポインタを記憶するのに利用されると好ましい。この方法は、大きいボリュームの当該選択領域にズーミングしながら能動ボリュームを制限する簡単な方法を提供する。また、任意に形付けしたサブボリューム（ブロック・サイズの細分性で）のレンダリングも可能にする。その上、多くの小さいボリュームを収容しているシーンは、非常に効率よくレンダリングされ得る。それは、すべてのボリ

ュームが３Ｄメモリ・ユニットの中のどこにでも存在できるし、３Ｄメモリ・ユニットよりもむしろ、ルックアップ表だけを各ボリュームについて再ロードすれば良いからである。

【００５３】

透視投影および／またはディテール・レベル（ＬＯＤ）を実施する１つ方法は、ｘ、ｙ方向における２倍スーパーサンプリングに依存する。したがって、トリリニア補間用の補間ユニットならびに勾配計算用の勾配評価ユニットの４倍の複製を使用すると好ましい。その結果、ＳＤＲＡＭ、Ｃｕｂｅ－５パイプライン間のデータパスはほぼ不変である。しかしながら、Ｃｕｂｅ－５パイプライン間のバンド幅は、オンチップ・スループットおよびバッファと同様に、４倍になる。その主たる理由は、ノーマル・モードの各サンプルが最高４つのサンプルと交換されるからである（すなわち、ｘ方向において倍速、ｙ方向において倍速）。

【００５４】

好ましくは、異方性データセットの取り扱いおよびスーパーサンプリングには、不透明度αの修正を行うとよい。結合された機能は、$\alpha' = 1 - \alpha)^{d/k}$、ここで、スーパーサンプリング・ファクタｋは、１ボクセル・セル当たりのサンプルの数を表しており、ｄは、各ボクセル・セルを通してサイト・レイが移動する距離（すなわち、サイト・レイの経路長）を表している。好ましくは、レンダリング中に、α′の急速ルックアップのために、ルックアップ表（ＬＵＴ）を使用する。

【００５５】

図５を続けて参照して、下層のボリューム・データセットの均一なサンプリングで終わるボリュメトリック・データの透視レンダリングでは、レベル間でフィルタリングを行いながら合成バッファ７４の再スケーリングが必要である。ディテール・レベル（ＬＯＤ）透視レンダリングでは、レベル間での合成バッファ７４の再アラインメントが必要である。これらのプロセス（パイプライン５２で利用できない大域的連絡を取り入れている）は、共に、好ましくは、ワープ単位（単数または複数）４４によって実施される。合成バッファ７４は既にワープ単位４４にアクセス可能となっているが、フィードバック・ライン４３を用いて合成

バッファ７４へ濾過された値を書き戻すと好ましい。

【００５６】

リアルタイム（すなわち、３０ヘルツのフレーム率）で最終的なフル・スクリーン画像を得るのにハードウェア・ワープ単位が必要である。図５に示すように、ベースプレーン画像（Ｃｕｂｅ－５レンダリング・パイプライン５２の合成ユニット６０によって生成した画像）は、２Ｄメモリ・ユニット４０においてバッファされると好ましい。２Ｄメモリ・ユニット４０からワープ単位４４までのメモリ・バンド幅を下げるために、ベースプレーン画像の各ピクセルは、一度だけアクセスされると好ましい。現および前のスキャンラインのサンプル間のリニア補間を実施するために、前のスキャンライン・サンプルを記憶するには、少なくとも１つのスキャンラインを保持するようなサイズとした別のＦＩＦＯユニットが必要である。各グリッド・ピクセルについての補間重みは、ホスト・マシンで事前計算すると好ましい。

【００５７】

不透明なジオメトリック・オブジェクトについてボリュームおよびジオメトリを正確に混ぜるために、Ｚバッファ画像は、合成バッファ６０に書き込まれると好ましい。合成ユニット６０は、各新しいサンプルを混合する前にｚ比較を実施しなければならない。その上、半透明のジオメトリック・スライスについては、ジオメトリ・エンジン７６は、好ましくは、本発明のジオメトリ入力バス（図４の参照コード４６）を利用して、ＲＧＢα値の各スラブをデータ流に挿入し、その結果、各スラブがボリュメトリック・データ・スライスでインタリーブされる。

【００５８】

テクスチャ・マッピングについて、たとえば、図６が、本発明に従って、３２ビットのテクセル・データを３Ｄメモリ・ユニットの１６ビット・ポクセルのミニブロックにおける２×２近辺にどのように記憶させられるかを示している。したがって、３２ビット・テクセルの４テクセル近辺が、各メモリ・バースト読み込み中に読み込まれると好ましい。データ重複がない場合、いくつかのテクスチャ座標が記憶済みのミニブロック間に位置し得るので、平均２．２５データ・バ

ースト読み取りで、適切なテクセル近辺にアクセスするようにＣｕｂｅ－５システムが実施すると好ましい。

【００５９】

再び図５を参照して、本発明の１つの形態に従って、ハードウェアで画像ベースのレンダリングを実施する１つ方法では、メモリ制御機構７８を利用する。この機構は、各Ｃｕｂｅ－５パイプライン５２に内蔵され、各パイプラインについての寄与領域に基づいて適切なソース・ピクセルを読み込むと好ましい。このパイプライン５２における補間ユニット（たとえば、５４、７２）が、次に、明視野レンダリングまたはルミグラフにとって必要とされる４次元（４Ｄ）補間を実施すると好ましい。別の実施例として、ワープ単位４４を利用してこの機能を実施してもよい。現視野に寄与しているソース・ピクセルが、好ましくはコネクション・ライン４１を通して２Ｄメモリ・ユニット４０に読み込まれ、組み立てられ、その後、ワープ変換が行われる。好ましくは、４つの組み立てられたソース画像は、４つの連続したワープ・パスにおいて処理される。４つの中間ワープド画像の最終的な組み合わせは、Ｃｕｂｅ－５パイプライン５２で実施される。先に説明したように、画像ベースのレンダリング・フィードバック・ライン６８は、中間ワープド画像を３Ｄメモリ２４に書き込むためのフィードバックを提供する。いずれの方法についても、３Ｄメモリ・ユニット２４は、大きい画像データベースのための局所的記憶装置を提供する。

【００６０】

ここで、上記の（Ｃｕｂｅ－５と呼ぶ）本発明の装置は、既に説明したユニバーサル・レンダリング以上に、普通のボリューム処理方法をかなり加速することができることは了解されたい。その上、本発明の装置は、性能を強化するために構成される多くのユニークなアルゴリズムと連動して使われてもよいＣｕｂｅ－５および／またはリアルタイム・ボリューム処理の性能を強化するおよび／またはリアルタイム・ボリューム処理のための強化した特徴を提供するようになっており、したがって、Ｃｕｂｅ－５システム全体を既存のボリューム・レンダリング・アーキテクチャ、たとえば、Ｃｕｂｅ－４より優れたものとすることができる多数のユニークなアルゴリズムと関連して使用することができる。画像ワーピ

ング、三次元変換、透視投影、大ボリューム処理、高品質レンダリング、クリッピング、デプス・キューイング、スーパーサンプリングおよび異方性データセットを実施するためのものを含むこれらユニークなアルゴリズムのいくつかを以下に詳しく説明する。

【００６１】

本発明の１つの形態によれば、画像ワーピングを実施する方法が提供される。この方法は、他の利点のうちでも、透視ワーピングの速度を上げ、向上する画質を提供する。画像ワーピングは、好ましくは、Ｃｕｂｅ－５ボリューム・レンダリング・パイプラインの最終ステージである。簡単に言えば、画像ワーピングは、主として、２つの画像、すなわち、ソース画像と目標画像の間のジオメトリック変換に関する。ジオメトリック変換は、ソース・ピクセルと目標ピクセルとの関係を定める。効率および高品質は、このような用途で等しく重要な問題である。本発明の装置においては、ワープ単位は、好ましくは、画像変換機能を実施する。その結果、ワープ単位を使用する用途は、本発明の画像ワーピング方法から利益を得る。

【００６２】

変換のデータフローによって区別すれば、画像ワーピング方法は、一般に、順方向ワーピングあるいは逆方向ワーピングのいずれかとして分類される。順方向ワーピングにおいては、ソース・ピクセルがスキャンライン・オーダーで処理され、結果が目標画像上へ投影される。逆方向ワーピングにおいては、ラスタ・オーダーにおける目標ピクセルが、逆にソース画像にマッピングされ、したがって、サンプリングされる。大部分の公知従来技術ワーピング・アルゴリズムは、逆方向ワーピングを使用する。

【００６３】

アフィン変換（すなわち、並進、回転、スケーリング、シャーリング等）と比較した場合、透視変換は、より高価で挑戦するに値すると考えられる。透視投影の場合、投影プレーンにおける１つのピクセルについてベースプレーン画像におけるサンプル場所を計算するときに、高価な除算が必要である。ＣＰＵによって実施するときには、普通の透視ワーピングは、代表的には、平行ワーピングより

も少なくとも三倍遅くなる。したがって、いくつかの従来技術方法は、透視変換を複数のパスを必要とするいくつかの単純な変換に分解していた。しかしながら、マルチパス変換アルゴリズムに固有の１つの主要な問題は、２つ一次元の（１Ｄ）フィルタリング作業の組み合わせが、真の二次元（２Ｄ）フィルタリングほぼ融通性がないということである。さらに、普通のマルチパス方法は、画質を劣化させる付加的なフィルタリング作業を行う。

【００６４】

本発明は、好ましくは、実質的にアフィン変換と同じ効率で実施することができるユニークな単パス順方向ワーピング方法を使用する。すべてのピクセルに対して伝統劇に実施されてきた高価な除算は、本発明によれば、１スキャンライン当たりたったの二回に減らされる。したがって、除算作業の数を減らすことによって、本発明は、たとえば、少なくとも速度および効果的なハードウェア実施に関して従来公知の技術による方法よりも優れた、代わりとなり得る透視ワーピング方法を提供する。本発明による、透視ワーピングのための好ましい方法を以下に説明する。

【００６５】

好ましくは、本発明は、透視ワーピングを実施するのにスキャンライン方法を使用する。しかしながら、ノーマル・ラスタ・スキャンライン・オーダーにおいて走査するよりもむしろ、本発明のアルゴリズムは、ソース画像における特殊なスキャンライン方向において処理される。図７、８に示すように、この特殊なスキャンライン方向９２（図８）は、好ましくは、ソース画像８０の平行スキャンライン８４が、目標画像８２における平行スキャンライン８６として現れるという特性を有する。そして、ソース・スキャンライン８４に沿った等距離サンプル・ポイント８８が、目標スキャンライン８６においても等距離サンプル・ポイント９０として残るという特性を有する。このユニークな方法のいくつかの利点としては、透視・補正画像ワーピングの複雑さの低減（すなわち、１ピクセルあたりの除算をなくし、１スキャンラインあたり２回の除算に代えることによる）、異方性フィルタリングを取り入れることによる正確なアンチエイリアシング、バイリニア補間によって生じたグーロー・シェーディングにおける欠点の訂正およ

び各ソース・ピクセルを正確に一度だけ読み込むことによるメモリ・バンド幅の最適化がある。

【００６６】

図８に示すように、この特殊なスキャンライン方向の直観は、投影ジオメトリに由来する。図８を参照して、ソース画像８０は、好ましくは、三次元（３Ｄ）表面に置かれ、そして、目標画像８２は、スクリーン上に置かれる。代表的なテクスチャ・マッピングと同様に、スクリーン上にピクセルを得るために、サイト・レイ（単数または複数）９４が、視点（またはアイ・ポイント）から３Ｄスペースにキャスティングされ、スクリーン８２および３Ｄ表面８０と交差させられる。交差ポイントが、サンプル・ポイント９８である。スクリーン・スペースにおける走査方向９２が３Ｄ平面と平行であるとき、両方の画像のスキャンラインは互いに平行である。そして、スキャンラインに沿った等距離のサンプル・ポイント９８は３Ｄ表面プレーンにおいても等距離のままである。この平行・保存（ＰＰ）スキャンライン方向が、所与の透視変換のために存在し、ユニークである。平行投影の場合、任意の走査方向が、両方の画像上にこの平行性を保存し、その簡略性によりラスタ・スキャンライン方向を使用するのが好ましいかも知れないことは了解されたい。

【００６７】

再び図７を参照して、平行・保存（ＰＰ）スキャンライン８４、８６は、それぞれ、ソース画像８０、目標画像８２の両方において示してある。ひとたび平行性を正しく達成したならば、ピクセル・アクセスは正規となり、そして、空間干渉性を両方の画像において利用することができる。その上、ＰＰスキャンラインは、ソース・サンプル８８の投影を算出するために各スキャンラインに対して除算を行うことなく、純粋増分アルゴリズムの適用を可能にする。しかしながら、非リニア投影によるすべてのスキャンラインの２つのエンドポイントのために、なお一回の除算は必要である。

【００６８】

図７を続けて参照して、ソース画像８０を、ラスタ方向よりもむしろＰＰスキャンライン方向において走査したとき、目標スキャンライン８６上のサンプル・

ポイント９０は、必ずしも目標ピクセル９１と一致する必要はない。しかしながら、サンプル・ポイント９０は、目標画像８２のｘグリッドライン８９上で整合させることができ、したがって、サンプル・ポイント９０は、ｙグリッド・ライン８７からのみ外れている（スキャンラインに沿って等距離である）。より効果的ではあるが、品質が低い実施例の場合、半ピクセルが最大エラーであるから、最も近い隣接の目標ピクセルにサンプル値を置くことが合理的近似である。しかしながら、より高い品質が好ましい場合には、本発明は、以下に説明するピクセル訂正および効果的アンチエイリアシングを実施することができる。

【００６９】

一般に、０（n^2）から０（ｎ）への除算の回数の低減は、本発明のアルゴリズムによって得られる（ここで、ｎは、リニア解像度である）。本アルゴリズムの場合、好ましくは、各サンプル・ポイントを算出するためには２回だけの加算が必要であるが、普通のラスタ・スキャンライン・アルゴリズムでは、一般に、１ピクセル当たり３回の加算、１回の除算および２回の乗算が必要である。本発明による、順方向画像ワーピングを実施する好ましい方法を以下に詳しく説明する。

【００７０】

本発明の順方向ワーピング・アルゴリズムは、好ましくは、２つのステージにおいて実施される。すなわち、（１）特殊な平行・保存（ＰＰ）スキャンライン方向を算出すること、そして、（２）各スキャンライン内で増分状態で、特殊なＰＰスキャンラインに沿って目標画像にソース画像を順方向マッピングすることである。

【００７１】

簡単に上述したように、平行・保存（ＰＰ）スキャンラインは、三次元（３Ｄ）平面とスクリーン（すなわち、目標画像）との間の交差ラインである。しかしながら、二次元（２Ｄ）問題においては、ＰＰスキャンラインは、２Ｄマトリックスに基づいて算出しなければならない。一般に、透視変換は、次のように表すことができる。

【００７２】

【数１】

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

ここで、（u、v）は、ソース・ピクセルの座標であり、（ｘ、ｙ）は、目標ピクセルの座標であり、Ｍは、透視変換マトリックスでる。（u、v）座標は（ｘ、ｙ）によって次のように表現することができる。

【００７３】

【数２】

$$(u,v) = F(x,y) = C \begin{bmatrix} ax + dy + g \\ bx + ey + h \end{bmatrix}$$

ここで、

【００７４】

【数３】

$$C = \frac{1}{(cx + fy + 1)}$$

目標画像におけるラインは、ｙ=ｋｘ+Ｂとして表現することができる。ここで、勾配ｋはライン方向を示し、Ｂはライン切片を示している。ＰＰスキャンラインについての勾配ｋを算出するためには、好ましくは、０、１の同じ勾配ｋおよび切片Ｂを有する２つの平行ラインを定義する。これらは、それぞれ、（０，０）、（１，ｋ）、（０，１）、（１，ｋ＋１）のポイント・ペアによって表される。ソース画像におけるこれらのポイントの座標を次に算出する。透視変換が直線を保存するので、これら２つのラインは、ソース画像における直線として残り、それらの勾配は、２つのポイント・ペアから算出することができる。ここで、

２のマッピングされたラインの勾配がほぼ同等であると仮定するならば、ｋにおける式が好ましくは得られる。ｋについてのこの式を解いた結果、

【００７５】

【数４】

$$k = -\frac{c}{f}$$

となる。ソース画像における対応する勾配ｋ′は、次の通りである。

【００７６】

【数５】

$$k' = \frac{bf - ec}{af - dc}$$

上記の式からわかるように、ｋ′＝－ｃ／ｆのとき、同種座標の分母は、Ｂｆ＋１の一定値となる。ここで、Ｂは、ｙ＝ｋｘ＋Ｂにおける切片である。

【００７７】

本発明の好ましい順方向ワーピング方法の第２ステージでは、スキャンライン処理を行う。これは、図９、１０に例示してある。次に図９を参照して、好ましいアルゴリズムは、ソース画像８０を通してスキャンライン８４（たとえば、スキャンラインＳ１～Ｓ４）を走査する。上記のように、スキャンライン８４は勾配ｋ′を有する。各スキャンライン８４に沿ったサンプル８８は、好ましくは、増分算出される。各スキャンライン８４について、まず、目標画像からソース画像上へエンドポイントの投影を算出する。次いで、スキャンライン上のサンプル・ポイントの数に基づいて、ｘ、ｙ両方向における増分を算出する。

【００７８】

ソース画像におけるサンプルの伝統的なバイリニア補間を考慮した場合、あらゆるサンプルは、本質的に、４つの周囲ソース・ピクセルの寄与を必要とする。ピクセルがすべてのサンプルについて毎回読み込まれる場合、各ピクセルは４回

読み込まれるべきである。これは、目標画像サイズの４倍であるメモリ・バンド幅に通じる。しかしながら、すべてのスキャンラインが平行であるから、隣接したスキャンライン上のサンプルは、通常、寄与しているソース・ピクセルを共有する。したがって、本発明の方法によれば、先に読み込まれていたピクセルは、好ましくは、共通ピクセルがソース画像自体からよりもむしろバッファから読み込まれるようにバッファされる。

【００７９】

図７を参照して、ピクセルは、好ましくは、ピクセル読み込みテンプレート１００と呼ばれる固定パターンにおいて読み込まれ、Bresenhamライン・アルゴリズム（当業者によって公知である）に基づいて算出される。図７の底に示している二進数は、読み込みテンプレート１００をコード化する１つ方法を表している。しかしながら、当業者によって明らかなように、本発明は、他のコード化方式も意図している。図７に示すように、このコードは、正のｖ方向において増加を示し、「０」は増加を表さず、「１」は１単位ずつの増加を示している。一方、ｕは、常に１単位ずつの増加を示している。図７の例の場合、ｕ軸線は、好ましくは、主要処理軸線と呼ぶとよいかも知れない。テンプレート１００が常に最も左のピクセルから出発し、垂直方向（すなわち、増加するｖ方向）に移動し、すべてのピクセルが、サンプリングにおける引き続く使用のためにバッファに読み込まれ、そこに置かれると好ましい。２つの点線間の任意のスキャンライン上でのサンプリングのためのピクセルを得るために、たとえ１つの特定のスキャンラインについても４つのピクセル・テンプレートが必要であるが、３つのピクセル・テンプレートだけでも充分であるかも知れない（たとえば、現スキャンラインＳ２を処理するのにテンプレート２、３、４のみが必要であるということ）ということは図７からわかるであろう。したがって、バッファ・サイズは、好ましくは、４本のスキャンラインである。

【００８０】

次に図１０Ａを参照して、ここには、バッファにおけるサンプルのアドレス指定が示してある。テンプレート・コード値が１であるときはいつでも、サンプルはｖ方向において１単位ずつ減少する。太いジグザクのライン１０４は、バッフ

ァにおける出力スキャンラインを表している。サンプルが陰影領域１０６内に入ると（この場合、バッファ内のピクセルがシャーされる）、サンプリングについての正しいピクセルを読み込むことに注意しなければならない。図１０Ｂは、この領域におけるサンプルｓの１つをバイリニア的に補間するための好ましい手順を示している。

【００８１】

バッファの内容は、好ましくは、スキャンライン位置に基づいて更新される。たとえば、図９を参照して、スキャンラインＳ１を処理するとき、テンプレート１、２、３、４がバッファ内にあることが好ましい。スキャンラインＳ２については、バッファは、好ましくは、同じままである。スキャンラインＳ３については、テンプレート５は、好ましくは、バッファ内に読み込まれ、テンプレート１は廃棄される。スキャンラインＳ４については、テンプレート６は、好ましくは、テンプレート２を交替する。これ以降は同じである。

【００８２】

上述したように、本発明のユニークな順方向画像ワーピング方法における特徴の１つは、グーロー・シェーディングにおける欠点の訂正である。グーロー・シェーディングは、ジオメトリック・オブジェクトの表面に陰影を付けるために用いられるポピュラーな強度補間アルゴリズムである。頂点のところにのみカラーを与えた場合、グーロー・シェーディングは、ラスタ・スキャンライン・オーダーでジオメトリの全ラスタライゼーションのために強度をバイリニア的に補間する。グーロー・シェーディング方法における欠陥は、この技術分野において周知であり、たとえば、Digital Image Warping, by G. Wolberg, IEEE Computer Society Press, 1990のような論文の主題となっている。

【００８３】

グーロー方法と関連した問題の１つは、斜めのライン（一例として）が透視投影のためにリニアにマッピングされないということである。斜めのラインが３Ｄスクリーン・スペース上へ透視投影されたとき、グーロー・シェーディングは、この斜めのラインを曲線に変換し、この曲線が、透視変換におけるラインを保存する特性を損なうということである。

【００８４】

本発明の画像ワーピング方法は、グーロー・シェーディングにおける透視ゆがみを補正するのである。スクリーン・スペースにおけるラスタに沿ったリニア補間が、ジオメトリック座標に変換されたときに、一般的に非リニアとなるために、透視ゆがみが存在する。しかしながら、本発明の特殊な走査方向を使用した場合、直線性はマッピングすることによって保存される。したがって、画像スペースおよびジオメトリック・スペースの両方において補間がリニアとなり、それによって、グーロー・シェーディングのゆがみを固定する。ここで、エッジに沿った補間がなお非リニアであり、したがって、スキャンライン・エンドポイントが正しい補間についてジオメトリック・スペースに変換されなければならないということは明らかであろう。

【００８５】

本発明の順方向マッピング・アルゴリズムは、最も近い隣接の近似をもって、伝統的な方法を使用して生成した画像からほぼ見分けがつかない目標画像を生成するという点で好ましい。しかしながら、より高い画質が望まれる場合には、本発明の方法は、好ましくは、正確なグリッド・ポイントでピクセル値を算出することができる。単純な目標ピクセル訂正方式は、好ましくは、この訂正を実施するように導入するとよい。

【００８６】

次に図１１を参照して、目標画像８２におけるサンプル・ポイント９０が整数のｘ座標上で整合していると仮定した場合、正確なピクセル・グリッド場所９１でピクセル値を得るためには、各ピクセルの直ぐ上、下の２つのサンプルのリニア補間が実施されると好ましい。このリニア補間を単に二回目のパスとして実施すると、サンプルを再度読み込まれなければならないので、コストが増大する可能性がある。その代わりに、各サンプルが生成される毎に、本発明の好ましい方法は、中間バッファリングをまったく行うことなく、各サンプルの、対応する上下のピクセルへの寄与率を拡張する。

【００８７】

図１１の例で示すように、より太い傾斜したスキャンライン１０８上に位置す

るサンプル１１２は、それらに隣接する陰影付きのピクセルに寄与する（スキャンライン上方ではより明るいシェーディングを、スキャンライン下方ではより暗いシェーディングを行う）。矢印は、各サンプル１１２が、好ましくは、２つのピクセルに寄与することを示している。両方の寄与率が集められるまで、ピクセルが書き出されないことが好ましい。したがって、１つのスキャンライン・バッファは、好ましくは、中間ピクセル値を記憶するために内蔵される。

【００８８】

正しくかつ能率的にピクセルを書き出すために、ピクセル書き込みパターン（ピクセル書き込みテンプレート１１０と呼ばれる）は、好ましくは、事前計算される。ピクセル読み込みテンプレート（たとえば、図９の参照コード１００）と異なり、ピクセル書き込みテンプレート１１０は、好ましくは、１つのスキャンラインに沿ってサンプル値のｙ座標値を切り捨てることによって、算出される。テンプレート１１０は、一連の整数ｙステップおよび真のスキャンライン８６からの部分的な距離ｄｙとしてコード化されると好ましい。最終的なリニア補間のために使用される重みは、それぞれ、上下のピクセルについてｄｙおよび１－ｄｙである。すべてのスキャンラインが垂直方向（すなわちｙ方向）において１単位離れていると好ましいので、テンプレートは１投影当たりたった一度だけ算出される。

【００８９】

本発明の順方向ワーピング方法は、アンチエイリアシングによって画質をさらに向上させることができる。平行・保存（ＰＰ）スキャンラインを使用することで、より高い品質でより安価なアンチエイリアシング方法を達成することができる。

【００９０】

再び図７を参照して、ソース画像の上方スキャンライン上にあるサンプル・ポイントは、下方スキャンライン上にあるよりもまばらであり、アンダーサンプリングからノーマルサンプリングへの移行部を生じている。したがって、適切な再サンプリング・フィルタを使用して上方スキャンラインについてのエイリアシングを避けることができる。等方性フィルタリングは、明らかに、不正確でぼんや

りした画像を生じさせる。異方性フィルタの必要性は、Survey of Texture Mapping, by P. S. Heckbert, IEEE Computer Graphics and Applications, 6 (11) : 56-67, November 1986のような論文、もっと近いところでは、Texram: Smart Memory for Texturing, by A. Schilling, et al., IEEE Computer Graphics and Applications, 16 (3) :32-41, May 1996に示されている。

【００９１】

各フィルタがそのフットプリントおよびプロファイルによって定められることは、当業者にとって公知である。目標サンプルを１つの円として仮定すると、ソース画像におけるその投影画像はそのフットプリントである。図１２に示すように、このフットプリント１１４は、一般に、円形（すなわち、等方性）であってはならないし、正方形（すなわち、ミップ・マッピングと同様の形）であってはならず、円錐形でなければならない。フィルタのプロファイルは、フットプリント内の寄与しているピクセルの重みを決定する。ｓｉｎｃフィルタが最適であるが、ガウス・フィルタは実施するのがより容易であり、その有限のフットプリントおよび良好なローパス特性のために好ましい。本発明の透視ワーピング・アルゴリズムは、異方性フットプリントを算出する際により高い精度を与え、より低いコストでより高い画質を生じさせる。

【００９２】

異方性フットプリントを算出する普通の方法を使用する場合、楕円の主軸を、すべてのピクセルについて算出しなければならない。近似値が提案されたけれども、これは高価な計算を残し、公知の増分方法を利用できないのである。これらの従来技術方法を使用して楕円の主軸を得るためには、ヤコビ行列式を計算しなければならない。しかしながら、本発明の画像ワーピング方法を使用すると、ヤコビ行列式の計算は除いてもよい。

【００９３】

本発明に従って異方性フットプリントを計算する好ましい方法に対する洞察を獲得するために、ヤコビ行列式の特性を、まず、分析する。ｘｙ目標画像からｕｖソース画像への一般化した逆方向マッピングは、以下のように先に定義した。

【００９４】

【数６】

$$\begin{bmatrix} u \\ v \end{bmatrix} = F(x,y) = C \begin{bmatrix} ax+dy+g \\ bx+ey+h \end{bmatrix}$$

ここで、

【００９５】

【数７】

$$C = \frac{1}{(cx+fy+1)}$$

一般化変換のためのヤコビ行列式Ｊは、ｘおよびｙの非リニア関数である。

【００９６】

【数８】

$$J = C^3 \begin{bmatrix} y(af-cd)+a-gc & x(af-cd)-d+gf \\ y(bf-ce)+b-hc & x(bf-ce)-e+hf \end{bmatrix}$$

普通のアンチエイリアシング方法では、ヤコビ行列式は、ソース画像における各ピクセルのフットプリントを決定するのに使用され、異方性フィルタリングにとって必要である。ｘｙラスタ・スペースにおけるスクリーン・ピクセル間の差は、［１，０］、［０，１］方向において方向性導関数を計算することによってソース画像に投影される。ソース画像スペースにおけるこれらの導関数は、ｒ１、ｒ２と呼ばれ、次のように定義される。

【００９７】

【数９】

$$r_1 = J\begin{bmatrix}1\\0\end{bmatrix} = C^{-2}\begin{bmatrix}y(af-cd)+a-gc\\y(bf-ce)+b-hc\end{bmatrix}$$

および

【００９８】

【数１０】

$$r_2 = J\begin{bmatrix}0\\1\end{bmatrix} = C^{-2}\begin{bmatrix}x(af-cd)-d+gf\\x(bf-ce)-e+hf\end{bmatrix}$$

これらのベクトル、r_1およびr_2、は、フットプリント１１４を近似する楕円の境界ボックスを定める。代表的には、これらのベクトル１１６、１１８は、普通の異方性フィルタリング方法（たとえば、楕円加重平均（EWA）、フットプリント組立体）のために、必要に応じて、すべてのピクセルについて算出される。これは、Cを計算するために１ピクセル当たりもう一回の除算を必要とする。本発明によれば、以下に説明するように、フットプリントを決定するためのより正確な方法が与えられる。

【００９９】

ヤコビ行列式が非リニア・マッピングのリニア近似であるから、それはより正確であり、したがって、ソース画像スペースにおける隣接したサンプルまでの距離を取り出すことによってフットプリントを計算するのに好ましい。隣接したサンプルの投影が既に計算されているので、本発明のこの方法は、付加的な除算をなんら必要としない。

【０１００】

平行・保存（ＰＰ）走査方向は、ヤコビ行列式を計算するのに、より大きい干渉性に備え、除算を与えない。ＰＰ走査オーダーにおける各ピクセルについて、フットプリントは、好ましくは、r_1'、r_2'によって定義される。ＰＰスキャンラインに沿った方向［１，ｋ］における方向性導関数r_1'は、

【０１０１】

【数１１】

$$r_1' = \nabla_{[1,k]} F = J \begin{bmatrix} 1 \\ k \end{bmatrix} = \dot{C}^2 \begin{bmatrix} af - cd \\ bf - ce \end{bmatrix}$$

であり、そして、y=kx+Bであるから、すべてのＰＰスキャンラインについてＣ＝１／（Ｂｆ＋１）は一定であり、したがって、r_1'もすべてのＰＰスキャンラインについて一定である。本発明の方法は、この事実を利用して、好ましくは、除算なしで、ソース画像座標を１つのスキャンラインに沿って増分させる。y方向［0，1］における方向性導関数r_2'の値は、

【０１０２】

【数１２】

$$r_2' = \nabla_{[0,1]} F = r_2$$

である。ここで、xの関数であり、したがって、スキャンラインに沿って増分させることができるので、r_2'がスキャンラインに沿って線形に変化することは明らかである。この特殊な走査方向は、ソース画像座標およびピクセル・フットプリントを簡単かつ能率的に計算するのを可能にする。

【０１０３】

すべてのフットプリントおよびソース・ピクセル座標情報を能率的に計算した後、正しい異方性フィルタリングを、当業者にとって公知の標準方法、たとえば、GreeneおよびHeckbertの楕円加重平均（ＥＷＡ）またはシリング等のフットプリント組立体を使用して実施することができる。これらの普通のアルゴリズムは、たとえば、テキストCreating Raster Omnimax Images from Multiple Perspective Views Using the Elliptical Weighted Average Filter, by N. Greene and P. S. Heckbert, IEEE Computer Graphics and Applications, 6 (6) :21-27, June 1986に記載されている。しかしながら、先に指摘したように、楕円フッ

トプリント近似値でさえ不正確であるから、これらの普通のフィルタリング方法は好ましくない。さらに、このような従来技術方法は、冗長なサンプリングを行う（すなわち、複数回、各ソース・ピクセルにアクセスする）ことになる。たとえば、１．０ソース・ピクセルのフットプリント半径を持つ円形フィルタ領域の場合、各ソース・ピクセルは、平均π回サンプリングされる。本発明の順方向マッピング技術を使用することによって、冗長なメモリ・アクセスが本質的に除去され、したがって、πの因数だけメモリ・バンド幅を下げることができる。好ましくは、本発明は、すべてのソース・ピクセルが、ピクセル読み込みテンプレート・オーダーで一度読み込まれ、次いで、フィルタ・カーネルで目標画像上へスプラットされる順方向マッピング技術を提供する。

【０１０４】

図１３に示すように、各ソース・ピクセル１２４は、その最も近い隣接の目標サンプル１２６の各々に対するΔｘ１２０およびΔｙ１２２を有する。Δｘは、好ましくは、スキャンラインに沿ったすべてのサンプルが等距離であるので、増分計算され得る。特殊な走査方向は、各スキャンラインに沿ってΔｙが一定となることをほぼ保証する。ラスタ・グリッド場所は真のスキャンライン１２８からずれているが、実際の距離は、ピクセル読み込みテンプレート１３０内に記憶され得る小訂正を加えることによって推定可能であり、好ましくはスキャンラインの中で均一である。フィルタ・カーネルは、好ましくは、一度事前計算され、ルックアップ表（ＬＵＴ）に記憶される。その後、各ソース・ピクセル１２４の寄与率が、４つ（またはそれより多くの）最も近い隣接の目標サンプル１２６についてルックアップ表（ＬＵＴ）にそのΔｘ、Δｙによって割り出されると好ましい。目標サンプル１２６の数は、使用するフィルタのフットプリントに依存する。そして、好ましくは、４から１６のサンプルまで変化してもよい。この方法を使用することで、各ソース・ピクセル１２４は、好ましくは、メモリから一度正確に呼び出され、次いで、４回（またはそれ以上）、ルックアップ表・エントリによって変調され、目標ピクセルに蓄積される。このようにして、最終的なピクセル値は、すぐ近くのソース・ピクセル１２４の加重平均となる。この加重平均は、各最終ピクセル強度を標準化するためにフィルタ重みの合計での除算を必要

とする。

【０１０５】

画像ワーピング（広義には２つの画像（たとえば、ソース画像および目標画像）間にジオメトリック変換として定義され得る）に加えて、三次元（３Ｄ）ボリューム変換は、ボリューム・レンダリング、ボリューム・モデリングおよび複数のボリュームの整合で重要な役割を演ずる。すべてのアフィン変換の中で、回転は、一般的に、最も多くの計算時間を費やし、最も複雑と考えられる。したがって、本発明に従ってユニバーサル３Ｄレンダリング・アーキテクチャを提供する際に、以下に詳しく説明するように、任意の３Ｄボリューム回転を実施するいくつかのユニークな方法が提起される。本発明のユニバーサル３Ｄレンダリング・ハードウェアはここに記載した３Ｄボリューム回転方法なしで使用され得るが、これらの方法またはアルゴリズムは、好ましくは、速度および特徴を向上させるために本発明の装置と一緒に実施され、本発明の装置を最も能率的に利用するようになっている。

【０１０６】

３Ｄボリューム回転を実施するユニークな方法を説明する前に、最初に使用する用語のいくつかの基本的な定義を与えることが重要である。当業者によって明らかなように、画像の行、列に対して、ボリューム内のビームとは、１つの主要座標軸線に沿ったボクセルの行と定義することができる（たとえば、ｘビームは、ｘ方向におけるボクセルの行である）。ボリュームのスライスとは、主要軸線に対して垂直であるボクセルのプレーンである（たとえば、ｘスライスは、ｘ軸線に対して垂直なプレーンとして定義される）。

【０１０７】

ボリューム変換を実施する従来技術方法は、代表的には、多パス・アルゴリズムを利用する。これらのアルゴリズムは、通常、画像変換のために使用される複数パス・アルゴリズムの直接的延長である。３Ｄ回転を実施する種々の方法が、提案されており、一般的に、複数の二次元（２Ｄ）変換あるいは一次元（１Ｄ）変換に３Ｄ変換を分解することを伴う。これらの従来技術方法は、Volume Rendering, by R. A. Drebin et al., Computer Graphics (SIGGRAPH '88 Proceedi

ngs），Vol. 22, pp 65-74, August 1988、Three-Pass Affine Transformations for Volume Rendering, by P. Hanrahan, Computer Graphics （San Diego Workshop on Volume Visualization），Vol. 24, pp 7 1-78, November 1990およびFast Rotation of Volume Data on Parallel Architectures, by P. Schroder and J. B. Salem, Visualization '91, pp. 50-57, 1991を含む論文の主題となっている。これらの論文は、すべて、参考資料としてここに援用する。しかしながら、これらの公知の３Ｄ変換方法は、代表的には、回転品質が低くおよび／または処理速度が遅くなる。

【０１０８】

三次元（３Ｄ）回転を難しくしている特性の１つは、３Ｄ回転が、本来、大域的連絡を必要としており、分散したメモリ・モジュールへデータを書き戻しながらメモリ・コンテンションを生じさせることにある。しかしながら、シャー変換が最も近い隣接コネクションを利用するので、本発明のユニークなアーキテクチャによって提供されるように、シャー変換それ自体は、極めて実行可能なマルチ・パイプライン式ハードウェア実装に役立つ。本発明は、さらに、任意の３Ｄ回転を実施する新規な方法、特に、３Ｄ回転を異なったタイプのシャー変換のシーケンスに分解することによる方法を提供する。

【０１０９】

普通の分解方法を用いる場合、２Ｄ回転を３つの一次元（１Ｄ）シャーに分解することができるので、３Ｄ回転に対する直接的な拡張では、９つの１Ｄシャーを必要とする。しかしながら、本発明によれば、以下に詳しく説明するように、任意の３Ｄボリューム回転についての好ましいシャー分解方法が４つ提供される。これらの方法とは、４パス２Ｄスライス・シャー、４パス２Ｄビーム・シャー、３パス・ビーム・スライス・シャーおよび２パス３Ｄビーム・シャー分解である。スケール作業を導入しないことによって、本発明のアルゴリズムは、サンプリング、フィルタリングおよびそれに伴う画像劣化における複雑化を避けることができる。

【０１１０】

ここで、当業者であれば、３Ｄ回転マトリックスを３つの主要軸線回転、R_x（

φ）、R_y（θ）、R_z（α）の連結として表すことができることは明らかであろう。ここで、

【０１１１】

【数１３】

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 0 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

である。この連結を実施するオーダーは、異なる３Ｄ回転マトリックスを生じさせる。全体で３Ｄ回転マトリックスの６つの順列がある。たとえば、下層の３Ｄ回転マトリックスは、$R3_D=R_x$（φ）R_y（θ）R_z（α）として選択された。個々で、

【０１１２】

【数１４】

$$R_{3D} = \begin{bmatrix} \cos\theta\cos\alpha & \cos\theta\sin\alpha & -\sin\theta \\ \sin\phi\sin\theta\cos\alpha - \cos\phi\sin\alpha & \sin\phi\sin\theta\sin\alpha + \cos\phi\cos\alpha & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\alpha + \sin\phi\sin\alpha & \cos\phi\sin\theta\sin\alpha - \sin\phi\cos\alpha & \cos\phi\cos\theta \end{bmatrix}$$

後に続くすべての分解のために、上記の３Ｄ回転マトリックス（$R3_D$）が使用される。本発明のユニークな方法と他の普通の方法との間の主要な差の１つは

、本発明においては、シャー・シーケンスを得るために、分解を、複数の２Ｄ回転シーケンスよりもむしろ、３Ｄ回転マトリックスに直接適用するということである。ここで、本発明に従って実施されるシャー操作の任意のものについて、好ましいハードウェア手段としてバレル・シフタを使用できるが、他の任意の手段（たとえば、対数シフタ等）も同様に考えられることは明らかであろう。

【０１１３】

図１４に示すように、本発明の一実施例による、二次元（２Ｄ）スライス・シャー回転を実施する方法は、好ましくは、３Ｄ回転を一連の２Ｄスライス・シャーに分解することを含む。２Ｄスライス・シャーにおいては、ボリューム・スライス（すなわち、主投影軸線に沿った、任意２つの軸線に対して平行なボクセルのプレーン）は、単にそのプレーン内で変位されるだけである。スライスは、任意に任意の主要投影軸線に沿って採用することができる。たとえば、図１４はｙスライス・シャーを示している。２Ｄｙスライス・シャーは、好ましくは、次のように表現される。

【０１１４】

【数１５】

$$x = x + a \cdot y$$
$$z = z + b \cdot y$$

２Ｄｙスライス・シャーは、好ましくは、Ｓ（ｘｚ、ｙ、（ａ、ｂ））として書くことができる。これは、ｘ方向における量ａ１３２だけ、そして、ｚ方向における量ｂ１３４だけｙ軸線に沿ったシャーと解釈される。ａ、ｂは好ましくは共に定数であるが、ａ、ｂが関数も表すことも考えられる。２Ｄｘスライス・シャー、Ｓ（ｙｚ、ｘ、（ｃ、ｄ））と、２Ｄｚスライス・シャー、Ｓ（ｘｙ、ｚ、（ｅ、ｆ）も同様に定義される。図１４を参照して、実線１３６で表されるボリュームは、点線１３８によって定義されたボリュームのシャー結果である。

【０１１５】

直観的に、同じ軸線に沿った連続的なシャーは、整合シャーを生成する。たとえば、

【０１１６】

【数１６】

$$S(xz,y,(a,b))\cdot S(xz,y,(a',b'))$$
$$=\begin{bmatrix}1&0&0\\a&1&b\\0&0&1\end{bmatrix}\cdot\begin{bmatrix}1&0&0\\a'&1&b'\\0&0&1\end{bmatrix}$$
$$=\begin{bmatrix}1&0&0\\a+a'&1&b+b'\\0&0&1\end{bmatrix}$$

一般的な３Ｄマトリックスを２Ｄシャー・マトリックスから作り上げるために、シャー・プロダクトは、異なったシャーのプロダクト、すなわち、Ｓ（ｘｚ、ｘ、（ｃ、ｄ））、Ｓ（ｘｚ、ｙ、（ａ、ｂ））、Ｓ（ｘｙ、ｚ、（ｅ、ｆ））に制限することができる。しかしながら、これら３つのシャー・マトリックスのプロダクト・マトリックスは、現在のマトリックスにおける定数１により、なお一般的な形態となることはない。したがって、別のシャー・マトリックスは、好ましくは、連結される。ここで、この最終的なシャーは、第１のシャーと同じスライス・シャーである。これは、以下の６つのシャー・シーケンス順列を生じさせる。

【０１１７】

【数１７】

$$S(xz,y,(a,b))\cdot S(xy,z,(e,f))\cdot S(yz,x,(c,d))\cdot S(xz,y,(g,h))$$
$$S(xz,y,(a,b))\cdot S(yz,x,(c,d))\cdot S(xy,z,(e,f))\cdot S(xz,y,(g,h))$$
$$S(xy,z,(e,f))\cdot S(xz,y,(a,b))\cdot S(yz,x,(c,d))\cdot S(xy,z,(i,j))$$
$$S(xy,z,(e,f))\cdot S(yz,x,(c,d))\cdot S(xz,y,(a,b))\cdot S(xy,z,(i,j))$$
$$S(yz,x,(c,d))\cdot S(xz,y,(a,b))\cdot S(xy,z,(e,f))\cdot S(yz,x,(m,n))$$
$$S(yz,x,(c,d))\cdot S(xy,z,(e,f))\cdot S(xz,y,(a,b))\cdot S(yz,x,(m,n))$$

シャー・シーケンスの各々について、連続的シャー・マトリックスのプロダクト・マトリックスが、好ましくは、計算され、下層の３Ｄ回転マトリックスに等しくなるように設定される。たとえば、第１のシャー・シーケンスについては、上記のように、すなわち、S(xz,y,(a,b))・S(xy,z,(e,f))・S(yz,x,(c,d))・S(xz,y,(g,h))とする。

【０１１８】

【数１８】

$$R_{3D} = R_z(\phi)R_y(\theta)R_x(\alpha) = S(xz,y,(a,b))\cdot S(xy,z,(e,f))\cdot S(yz,x,(c,d))\cdot S(xz,y,(g,h))$$

上記のマトリックス式は、８つの変数、すなわち、ａ、ｂ、ｃ、ｄ、ｅ、ｆ、ｇ、ｈを持つ９つの三角法の式を意味する。８つの変数ａ～ｈについてこれら９つの式を解くと、次のような結果が得られる。

【０１１９】

【数１９】

$$a = \frac{\sin\theta\sin\alpha(\cos\theta - \cos\phi) + \sin\phi(\cos\alpha - \cos\theta)}{(\cos\theta)^2\sin\alpha\sin\phi}$$

$$b = -\frac{\cos\phi\cos\theta - 1}{\sin\phi\cos\theta}$$

$$c = \cos\theta\sin\alpha$$

$$d = \frac{-\sin\phi\sin\theta\cos\alpha + \cos\phi\sin\alpha - \cos\theta\sin\alpha}{\sin\phi\cos\theta}$$

$$e = \frac{\sin\phi\cos\theta + \cos\phi\sin\theta\sin\alpha - \sin\phi\cos\alpha}{\cos\theta\sin\alpha}$$

$$f = -\sin\phi\cos\theta$$

$$g = \frac{\cos\theta\cos\alpha - 1}{\cos\theta\sin\alpha}$$

$$h = \frac{\cos\theta\cos\alpha - 1}{\cos\theta\sin\alpha}$$

同様にして、上記の残っている５つのスライス・シャー・シーケンスについてのシャー・マトリックスを得ることができる。実際、上記の解を持つスライス・シャー・シーケンスは、最も単純な表現を有し、好ましくは、基本シーケンスと呼ぶ。

【０１２０】

次に図１５を参照して、二次元（２Ｄ）ビーム・シャー分解による三次元（３Ｄ）回転を実施する方法を以下に説明する。まず、或るビーム・シャーを、他の２つの座標をなんら変化させることなく、主方向において変位されただけであるビームと定義する。たとえば、２Ｄｘビーム・シャーは、好ましくは、

【０１２１】

【数２０】

$$x = x + a \cdot y + b \cdot z$$

と表現する。

【０１２２】

２Ｄｘビーム・シャーは、好ましくは、Ｓ（ｘ、ｙｚ、（ｃ、ｄ））と書くことができ、これは、ｘ方向における量ａおよびｚ方向における量ｂだけｘ軸線に沿ったシャーと解釈される。２Ｄｙビーム・シャー、Ｓ（ｙ、ｘｚ、（ａ、ｂ））、そして、２Ｄｚビーム・シャー、Ｓ（ｚ、ｘｙ、（ｅ、ｆ））も同様に定義する。図１５は、ｘビーム・シャーを示しており、そこにおいて、点線１４６で表されるボリュームは、実線１４４によって表されるボリューム位置にシャーされる。

【０１２３】

二次元（２Ｄ）ビーム・シャーは、２Ｄスライス・シャー以上に有利であり、したがって、好ましい。これは、１つのビームが他の２つの座標を変えずに変位されるからである。したがって、２Ｄビーム・シャー方法の各パスについて再サンプリングがより簡単になる。これは、リニア補間のみが必要高等である。対照的に、２Ｄスライス・シャー方法は、より複雑であるバイリニア補間を必要とする。

【０１２４】

２Ｄビーム・シャー・マトリックス分解と同様に、一般的な３Ｄマトリックスを２Ｄビーム・シャー・マトリックス分解から作り上げるために、シャー・プロダクトは、異なるシャーのプロダクト、すなわち、Ｓ（ｘ、ｙｚ、（ｃ、ｄ））、Ｓ（ｙ、ｘｚ、（ａ、ｂ））、Ｓ（ｚ、ｘｙ、（ｅ、ｆ））に制限することができる。しかしながら、これらの３つのシャー・マトリックスのプロダクト・マトリックスは、マトリックスにおける定数１により、なお一般的に形態とならない。したがって、スライス・シャー方法と同様に、好ましくは、別のシャー・マトリックスを連結する。ここで、この最終的なシャーは、第１シャーと同じビーム・シャーである。これは、以下の６つのシャー・シーケンス順列を生じさせる。

【０１２５】

【数２１】

$$S(y,xz,(a,b))S(z,xy,(e,f))S(x,yz,(c,d))S(y,xz,(g,h))$$
$$S(y,xz,(a,b))S(x,yz,(c,d))S(z,xy,(e,f))S(y,xz,(g,h))$$
$$S(z,xy,(e,f))S(y,xz,(a,b))S(x,yz,(c,d))S(z,xy,(i,j))$$
$$S(z,xy,(e,f))S(x,yz,(c,d))S(y,xz,(a,b))S(z,xy,(i,j))$$
$$S(x,yz,(c,d))S(y,xz,(a,b))S(z,xy,(e,f))S(x,yz,(m,n))$$
$$S(x,yz,(c,d))S(z,xy,(e,f))S(y,xz,(a,b))S(x,yz,(m,n))$$

上記シャー・シーケンスの各々について、連続的シャー・マトリックスのプロダクト・マトリックスが、好ましくは、計算され、下層の３Ｄ回転マトリックスに等しくなるように設定される。たとえば、第１のシャー・シーケンスについては、上記のように、すなわち、S(y,xz,(a,b))・S(z,xy,(e,f))・S(x,yz,(c,d))・S(y,xz,(g,h))とする。

【０１２６】

【数２２】

$xy, (e, f)) S(x, yz, (c, d)) S(y, xz, (g, h))$:

$$R_{3D} = R_z(\phi) R_y(\theta) R_z(\alpha) =$$
$$S(y, xz, (a, b)) \cdot S(z, xy, (e, f))$$
$$\cdot S(x, yz(c, d)) \cdot S(y, xz, (g, h))$$

上記のマトリックス式は、８つの変数、すなわち、ａ、ｂ、ｃ、ｄ、ｅ、ｆ、ｇ、ｈを持つ９つの三角法の式を意味する。８つの変数ａ～ｈについてこれら９つの式を解くと、次のような結果が得られる。

【０１２７】

【数２３】

$$a = \frac{\sin\theta \sin\alpha(\cos\phi - \cos\theta) + \sin\phi(\cos\theta - \cos\alpha)}{\sin\phi \sin\alpha(\cos\theta)^2}$$

$$b = -\frac{\cos\phi\cos\theta - 1}{\sin\phi\cos\theta}$$

$$c = -\cos\theta\sin\alpha$$

$$d = \frac{\cos\theta\sin\alpha + \sin\phi\sin\theta\cos\alpha - \cos\phi\sin\alpha}{\sin\phi\cos\theta}$$

$$e = -\frac{\cos\phi\sin\theta\sin\alpha - \sin\phi\cos\alpha + \sin\phi\cos\theta}{\cos\theta\sin\alpha}$$

$$f = \sin\phi\cos\theta$$

$$g = -\frac{\cos\theta\cos\alpha - 1}{\cos\theta\sin\alpha}$$

$$h = \frac{\sin\phi\sin\theta(\cos\theta - \cos\alpha) + \sin\alpha(\cos\phi - \cos\theta)}{\sin\phi\sin\alpha(\cos\theta)^2}$$

同様にして、上記の残りの５つのビーム・シャー・シーケンスについてのシャー・マトリックスを得ることができる。上記の解を持つビーム・シャー・シーケンスは、好ましくは、基本シーケンスと呼ぶ。

【０１２８】

次に図１６を参照して、本発明による、二次元（２Ｄ）ビーム・スライス・シ

ャー分解による三次元（３Ｄ）回転を実施する方法を説明する。２Ｄビーム・スライス・シャーは、好ましくは、１つのプレーン内で変位させられたビームと定義することができる。たとえば、２Ｄｘビームｙスライス・シャーは、好ましくは、

【０１２９】

【数２４】

$$x = x + a \cdot y + g \cdot z$$

$$z = z + b \cdot y$$

と表現される。

【０１３０】

２Ｄｘビームｙスライス・シャーは、好ましくは、Ｓ（（ｘ、ｙｚ、（ａ、ｇ））、（ｚ、ｙ、ｂ））と書くことができ、これは、ｙ方向において量ａ、ｚ方向において量ｇだけｘ軸線に沿ったシャーをｙ方向において量ｂだけｚ軸線に沿ったシャーと組み合わせたものと解釈される。ここで、ａ、ｇおよびｂは、好ましくは、定数である。本質的に、ビーム・スライス・シャーは、ビーム・シャーとスライス・シャーの組み合わせである。図１６は、ｘビームｙスライス・シャー、Ｓ（（ｘ、ｙｚ、（ａ、（ｇ））、（ｚ、ｙ、ｂ））を示しており、そこにおいては、点線１５６で表されるボリュームは、実線１５４によって表されるボリューム位置にシャーされる。

【０１３１】

一般的な３Ｄマトリックスを２Ｄシャー・マトリックス分解から作り上げるために、シャー・プロダクトは、異なったシャーのプロダクト、すなわち、ｙビームｘスライス・シャーＳ（（ｙ、ｘｚ、（ｃ、ｈ））、（ｚ、ｘ、ｄ）、ｘビームｙスライス・シャーＳ（（ｘ、ｙｚ、（ａ、ｇ））、（ｚ、ｙ、ｂ）およびｙビーム・シャーＳ（ｙ、ｘｚ、（Ｉ、ｆ））に制限するとよい。スライス・シャー、ビーム・シャー方法の場合と同様に、６つのビーム・スライス・シャー・シーケンス順列があることは了解されたい。

【０１３２】

シャー・シーケンスの各々について、連続的シャー・マトリックスのプロダクト・マトリックスが、好ましくは、計算され、下層の３Ｄ回転マトリックスに等しくなるように設定される。たとえば、上記の第１のシャー・シーケンスについては、S((y,xz,(c,h)),(z,x,d)) S(x,yz,(a,g)),(z,y,b)S(y,xz,(I,f))とする。

【０１３３】

【数２５】

$$(z,x,d))\,S((x,yz,(a,g)),(z,y,b))\,S(y,xz,(I,f))):$$

$$R_{3D} = R_x(\phi)\,R_y(\theta)\,R_x(\alpha) =$$

$$S((y,xz,(c,h)),(z,x,d))\cdot S((x,yz,(a,g)),(z,y,b))$$

$$\cdot S(y,xz,(I,f))$$

上記のマトリックス式は、８つの変数、すなわち、ａ、ｂ、ｃ、ｄ、ｆ、ｇ、ｈ，Ｉを持つ９つの三角法の式を意味する。８つの変数についてこれら９つの式を解くと、次のような結果が得られる。

【０１３４】

【数２６】

$$a = \sin\phi\sin\theta\cos\alpha - \cos\phi\sin\alpha$$

$$b = \sin\phi\cos\theta$$

$$c = \frac{\sin\phi(\cos\theta - \cos\alpha) + \sin\theta\sin\alpha(\cos\phi - \cos\theta)}{\sin\phi(\cos\theta)^2\sin\alpha}$$

$$d = \frac{\sin\phi\cos\alpha - \cos\phi\sin\theta\sin\alpha - \sin\phi\cos\theta}{\cos\theta\sin\alpha}$$

$$f = \frac{\sin\phi\sin\theta(\cos\theta - \cos\alpha) + \sin\alpha(\cos\phi - \cos\theta)}{\sin\phi(\cos\theta)^2\sin\alpha}$$

$$g = \frac{\cos\theta\sin\alpha + \sin\phi\sin\theta\cos\alpha - \cos\phi\sin\alpha}{\sin\phi\cos\theta}$$

$$h = \frac{\cos\phi\cos\theta - 1}{\sin\phi\cos\theta}$$

$$i = -\frac{\cos\theta\cos\alpha - 1}{\cos\theta\sin\alpha}$$

残りの５つのビーム・シャー・シーケンスについてのシャー・マトリックスを同様にして得ることができることは明らかである。

【０１３５】

図１７は、本発明に従って、３Ｄビーム・シャー分解を使用して任意の三次元（３Ｄ）回転を実施する第４の方法を示している。上記のビーム・スライス・シャー分解方法で使用した連続的シャー・マトリックスのプロダクト・マトリックス（すなわち、Ｓ（（ｙ、ｘｚ、（ｃ、ｈ））、（ｚ、ｘ、ｄ）・Ｓ（（ｘ、ｙｚ、（ａ、ｇ））、（ｚ、ｙ、ｂ）・Ｓ（ｙ、ｘｚ、（Ｉ、ｆ））をさらに点検することによって、２Ｄシャーの最初のペア、最後のペアを合併することができる。これは、各ペアには共通のビームがあるからである。たとえば、ｘビームは、第１ペアのｙスライス、ｚスライスのシャーの共通ビームである。したがって、シャーの数は、３Ｄビーム・シャーの新定義を導入することによって２まで減らすことができる。

【０１３６】

図１７は、３Ｄｘビーム・シャーを示している。これは、２つの連続した２Ｄスライス・シャーＳ（ｘｚ、ｙ、（ａ、ｂ））Ｓ（ｘｙ、ｚ、（ｅ、ｆ））の連結に等しい。ここで、他に２つの３Ｄビーム・シャーがある、すなわち、Ｓ（ｙｚ、ｘ、（ｃ、ｄ））Ｓ（ｘｚ、ｙ、（ａ、ｂ））と表される３Ｄｚビーム・シャーと、Ｓ（ｙｚ、ｘ、（ｃ、ｄ））Ｓ（ｘｙ、ｚ、（ｅ、ｆ））と表される３Ｄｙビーム・シャーとがあることは了解されたい。すべての３Ｄビーム・シャーは、好ましくは、たった１つの主ビームだけを含む。図１７を参照して、マークを付けたｘビーム１５８（暗い陰影のビーム）は、好ましくは、矢印をたどる新しい３Ｄ場所に並進させられる。明るい陰影を付けたビーム１５８′は、シャー分解を２つの連続的２Ｄスライス・シャーと解釈された場合の中間位置を示す。

【０１３７】

３つの３Ｄビーム・シャーは、好ましくは、SH_{Hx}、SH_{Hy} およびSH_{Hz} と示すとよい。ここで、ここに記載した本発明の方法を用いた場合、任意の３Ｄ回転を、ほんの２つの連続３Ｄビーム・シャーに分解できる。基本分解シーケンスは、以下のように２Ｄスライス・シャー・シーケンスから直接得ることができる。

【０１３８】

【数２７】

$$R_{3D} = SH_{3D_x} \cdot SH_{3D_z}$$

ここで、

【０１３９】

【数２８】

$$SH_{3D_x} = S(xz,y,(a,b)) \cdot S(xy,z,(e,f))$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ a+be & 1+bf & b \\ e & f & 1 \end{bmatrix}$$

$$SH_{3D_z} = S(yz,x,(c,d)) \cdot S(xz,y,(g,h))$$

$$= \begin{bmatrix} 1+cg & c & d+ch \\ g & 1 & h \\ 0 & 0 & 1 \end{bmatrix}$$

ここに説明した本発明の３Ｄビーム・シャー分解方法を使用すると、任意の３Ｄ回転は、好ましくは、２つの主要ビーム変換しか伴わないのに対して、普通の分解方法は、３つ（たとえば、Hanrahanの分解）を必要とする。本発明の３Ｄビーム・シャー方法によれば、第１のパスはｘビームのみを含み、第２のパスはｚビームのみを含む。第１のシャー・パスの終りまでに、ビームのすべてのボクセルは、好ましくは、同じオフセットを有する。N^3個のボリュームについてN^2個のビームがあるので、異なったオフセット値はN^2個だけある。したがって、N^2個のビームについてのオフセット値は、第１パスの終りで記憶することができると共に、最も近い隣接の一体位置にボクセルを記憶することができる。

【０１４０】

多パス・アルゴリズムが使用されるとき、選ばれた再サンプリング技術は、高

品質を達成するのに重要である。直観的に、連続するシャー変換がグリッド・ポイントを離れるようにボクセルを移動させることができるので、再サンプリングが各パスについて必要である。しかしながら、連続回転をボリュームに適用した場合、多数回の再サンプリングに固有の問題の１つは、ボリューム品質の急速な劣化にある。したがって、ボリュームを一度だけサンプリングすることが望ましい。

【０１４１】

したがって、本発明の好ましい方法は、ボリュームの１パス式再サンプリングを達成する。本質的に、本発明の方法は、サンプリングしたボリュームを事前計算し、次に、各シャー・パスにおいてゼロ・オーダー（たとえば、最も近い隣接の）補間を使用する。これは、大域的連絡を必要とする公知の従来技術方法（たとえば、Wittenbrink、Somaniの順列ワーピング）と異なる。

【０１４２】

或るオリジナルのボリューム（ソース・ボリューム）と所望の回転したボリューム（目標ボリューム）とを与えた場合、本発明の方法は、好ましくは、まず、ソース・ボクセルと目標ボクセルとの間の１対１の対応関係を構築する。この１対１のマッピングは、各シャーがゼロ・オーダー補間を使用した１対１変換であるから、本発明のマルチパス・シャー分解によって保証される。一連の１対１マッピングの連結は、１対１のままである。ひとたびこの１対１の対応関係が確立されたならば、本発明の方法は、好ましくは、各ソース・ボクセルについて、それに対応する目標ボクセルを計算し、それをソース・ボクセル位置に記憶する。この手順中、大域的連絡はまったく不要である。すなわち、再サンプリングが、局所的ボクセルについての補間によって実施される。各目標ボクセルのサンプリング位置は、好ましくは、逆方向回転変換を使用して計算する。

【０１４３】

すべての目標ボクセルについて値を得た後、本発明の方法は、好ましくは、目標ボリュームにおける宛先に目標ボクセルをシャッフルする。直観的に、これは大域的連絡を含む。しかしながら、並列実行のために行うには大域的連絡は高価である。したがって、本発明による方法は、好ましくは、最も近い隣接配置方式

と共に複数のシャーを使用し、このボクセル・シャッフリングを達成する。シャーが規則的な非対立変換であるから、各パスは、大域的連絡を利用する場合よりも能率的に実施され得る。ここに説明した本発明の３Ｄビーム・シャー分解方法を使用することで、実質的に大域的連絡と同じ効果を達成するのに、規則的な局所的連絡のパスを最低２回行うだけでよい。

【０１４４】

ここで、３Ｄビーム・シャーにおけるビームのオーバーラップを避けることに注意しなければならないことは明らかである。たとえば、ここで、上記の３Ｄｘビーム・シャー式を考える。各ｘビームを保存するとき（すなわち、ｘビームが３Ｄｘビーム・シャーの後に剛性のままとなる）、いくつかのｘビームが互いにオーバーラップする可能性がある。必要な１対１マッピングを維持するために、上述したように、３Ｄビーム・シャーが２つの２Ｄスライス・シャーの連結であることを想起されたい。ゼロ・オーダー補間を使用するとき、２Ｄスライス・シャーは１対１マッピングを維持する。したがって、解として、本発明の方法は、好ましくは、２つの連続した２Ｄスライス・シャーのオーダーと同じオーダーを用いて宛先座標を計算するが、連絡は一度だけ行われるのが好ましい。３Ｄｘビーム・シャーの場合、３Ｄシャー・マトリックス（上記）を用いてｘ座標を直接的に算出しながら、各ビームのｙ、ｚ座標を、次のように計算すると好ましい。

【０１４５】

【数２９】

$$z' = round(z + b \cdot y)$$

$$y' = round(y + f \cdot z')$$

ここで、round（ｗ）は、最も近い整数に対する丸め関数ｗである。座標（ｙ′、ｚ′）は、最も近い隣接のストレージについてのビーム全体の一体座量を決定する。こうして、オーバーラップが生じることはない。

【０１４６】

本発明の別の形態によれば、強化ボリューム処理を実施するいくつかのユニー

クな方法を以下に詳しく説明する。

【０１４７】

透視投影は、特にレイ・キャスティングを実行するときに、固有の問題を提起する。平行投影の場合、ボリューム・データセットを通してキャスティングされるサイト・レイは、下層のボリューム・データについて一定のサンプリング・レートを維持する。必要な品質の出力画像を創作するようにこのサンプリング・レートをセットするのは簡単である。しかしながら、透視投影の場合、レイは、このような連続して均一なサンプリング・レートを維持しない。その代わりに、レイは、前から後へボリュームを移動させるにつれて発散する。図１８ａ、１８ｂに示すように、これは、下層のボリュームに不均一なサンプリングを創り出す。

【０１４８】

次に図１８ａ、１８ｂを参照して、普通のビーム・キャスティング・アルゴリズムは、一般的に、２つの方法のうちの１つによって、透視投影からのレイ発散を取り扱う。第１の方法は、アンダーサンプリング（図１８Ａ）であり、そこにおいては、レイ１６０が所定の視点１６２からキャスティングされ、その結果、ボリューム１６４のフロントでのサンプリング・レートが所望の画質に対して適切なものとなる。しかしながら、透視レイ発散のために、下層のボリューム・データセットは、アンダーサンプリングされる。これは、ボクセルの領域がサンプリングされないままであるボリューム１６６のリアに「ホール」を創り出すことによって厳しいエイリアシングとなる可能性がある。第２の方法は、オーバーサンプリングであり（図１８Ｂ）。そこにおいて、レイ１６０が所定の視点１６２からキャスティングされ、その結果、ボリューム・データセット１６６のリアでのサンプリング・レートが、所望の画質にとって適切なものとなる。この方法では、第１の方法のエイリアシングを回避することができるが、ボリュームはフロント１６４において急進的にオーバーサンプリングされる可能性がある。ボリューム１６４のフロントにおける効率の悪いオーバーサンプリングは、この方法のランタイムを劇的に増大させる。レイ１６０は、アンダーサンプリングとオーバーサンプリングの間のサンプリング・レートでキャスティングされ得る。これは、オーバーサンプリングの画質とアンダーサンプリングのレンダリング速度との

間のトレードオフとなる。

【０１４９】

多くの従来の像形成アーキテクチャは、透視投影を実施しようという試みさえしていない。他のアーキテクチャは、所定の視点からの発散サイト・レイをキャスティングすることによって、透視投影を処理していた。これは、一時的なエイリアシングで画像を生成し、真のリアルタイム・フレーム率（すなわち、３０ヘルツ）を達成せず、本発明のスライス・オーダー方法よりもかなり複雑である。

【０１５０】

レイ分割方法は、適応スーパーサンプリングの概念を適用し、均一なレイ密度を維持する。この方法では、隣接したレイが或る種の所定閾値を越えて発散するとき、レイを２つのチャイルド・レイに分割する。最近になって、視点からの距離に基づいて視野切頭体を領域に分割する方法が提案された。その結果、各領域のレイ密度が下層ボリューム解像度に近くなるのである。その後、この方法では、サブ画像に各領域を投影し、テクスチャ・マッピング・ハードウェアを使用してフレーム・バッファにこれらのサブ画像を合成する。実際、この技術は、連続的なレイを１つの領域を通してキャスティングし、次いで、指定境界のところで、それらを新しいセットの連続レイに分割する。しかしながら、これは、領域間の潜在的な、望ましくない不連続性を創り出す。

【０１５１】

本発明の一形態による、ＥＲ透視（指数的な領域透視）と呼ばれる、均一な規則的データセットの透視投影を実施するための方法は、好ましくは、下層のボリュームを適切にサンプリングし、それによって、普通のボリューム・レンダリング・システム及び方法に固有の上記問題がほぼ除去される。ＥＲ透視アルゴリズムは、アンダーサンプリング、オーバーサンプリングの両方の望ましい特性を組み合わせ、アンダーサンプリング方法のオーダーについてのランタイムを与えながら、オーバーサンプリングと関連した極めて良好なアンチエイリアシング特性を提供する。さらに、このアルゴリズムは、好ましくは、ボリューム・データセットのすべての可視ボクセルについて少なくとも１つのサンプルを創り出す。ＥＲ透視は、公知の透視投影方法と比較して同等あるいはそれより良好な画質を維

持しながらスライス・オーダー・ボクセル・アクセスを利用することによって従来の技術より以上のランタイム利益を獲得する。

【０１５２】

図１９は、本発明によるＥＲ透視アルゴリズムの２Ｄ頂面図である。図１９に示すように、ＥＲ透視アルゴリズムは、好ましくは、所定の視点１７２からの主要投影軸線（たとえば、ｚ軸）に沿った指数的に増大する距離に基づいて視野切頭体１６８を複数の領域に分割することによって機能する。好ましくは、連続したサイト・レイ１７４がボリューム・データセットを通して後から前（または前から後）へ視点１７２からキャスティングされ、ひとたびレイ１７４が互いに接近しすぎ（あるいは離れすぎた）ならば、これらのレイ１７４を合併（あるいは、分割）する。ＥＲ透視アルゴリズムの動作が前から後へのレイ・キャスティングと比較して後から前に付いて類似しているので、ＥＲ透視アルゴリズムの残りの説明は、合併を伴う後から前へのレイ・キャスティングのより直観的なケースに限定する。これらの差異は、それらが意味を持つときに指摘する。

【０１５３】

ＥＲ透視アルゴリズムは、好ましくは、領域境界１７０を使用する。これは、指数領域を定め、サイト・レイ１７４が好ましくは合併される場所にマークを付ける。領域を定義して、境界１７０ですべてのレイ１７４を合併することによって、このアルゴリズムは、局所的近傍状況よりもむしろ大域的ジオメトリに依存する或る規則的なレイ合併パターンを提供する。図２０Ａは、特に、ベースプレーン・ピクセルＢに対する寄与率について領域境界１７０でサイト・レイを合併することをより明瞭に示している。図２０Ａを参照して、奇数のレイ１７４が、好ましくは、合併され、その結果、生じたレイ１７４′は、本質的に、先の中心レイの正確な連続レイとなり、したがって、領域境界１７０に存在する潜在的な不連続性を排除する。これは、公知従来方法以上の、本発明方法の重要な利点の１つである。さらに、このアルゴリズムは、最適状態でボリュームをサンプリングするときに達成されるフィルタリングを特徴づけることによって限定され得る。

【０１５４】

好ましいフィルタ方式の一例が、図２０Bに示してある。図２０Bを参照して、バートレット・ウィンドウ（すなわち、リニア補間、三角形フィルタ）を使用すると好ましい。各領域境界１７０で効率的な局所バートレット・ウィンドウをカスケーディングすることは、各ベースプレーン・ピクセル（図２０A参照）についての単一の大きなバートレット・フィルタでレイ１７４を再サンプリングすることとほぼ同等である。好ましいフィルタ重み１７５のグラフ図が、ベースプレーン・ピクセル（たとえば、ピクセルA、B、C）に対する寄与率について示してある。

【０１５５】

このアルゴリズムの基本的なサンプリング・レートは、所望の画質に従って所定の値にセットすることができる。基本サンプリング・レートは、下層のボリューム解像度と比較した最小限のビーム密度である。本発明のＥＲ透視方法は実質的に任意のサンプリング・レートをサポートするけれども、１ボクセル当たり少なくとも１つのレイのサンプリング・レートが好ましい。このアルゴリズムは、基本サンプリング・レートの一倍と二倍の間にレイ密度を維持するという利点を有する。このことは、ボクセルがボリューム・データセットの背後で失われることがないことを保証し、二倍のスーパーサンプリングで実行された総仕事量について上限を置く。

【０１５６】

本発明がスライス・オーダー処理を利用するので、ボリューム・データセットは、視野方向に対して最も垂直であるボリュームのベースプレーン上へ投影される。次に、ベースプレーン画像は、普通の方法（たとえば、シャー・ワープあるいは従来のＣｕｂｅ－４アーキテクチャと同じ方法）で最終的な画像プレーン上へワープされる。

【０１５７】

本発明のＥＲ透視方法は、上記のＣｕｂｅ－５アーキテクチャで実施するのに理想的である。具体的には、このアルゴリズムは、好ましくは、処理要素間の最も近い隣接連絡を必要とするだけである。処理要素の一次元アレイ上のボクセル行を処理しながら、このアルゴリズムは、処理要素がそれらの直ぐ左隣、右隣の

処理要素と連絡することを必要とするだけである。Ｃｕｂｅ－５レンダリング・パイプラインは、同様に、最も近い隣接連絡をサポートする。

【０１５８】

本発明のＥＲ透視アルゴリズムは、好ましくは、３つの主要軸線の１つに沿ったスライス・オーダー処理を使用する。したがって、ＥＲ透視アルゴリズムにおける領域は、主要投影軸線に沿ったスライスのスラブとして定義される。本発明のＥＲ透視方法の好ましい実施例においては、ボリューム・データセットは、ｚ軸に対して垂直なスライスに沿って投影される。ｚ軸のみに沿った投影に本発明の方法を限定するつもりはないが、座標系をフリップさせ、ジオメトリを回転させ得ることは了解されたい。本アルゴリズムは、図７に示すように、視点８６からボリューム・データセット８０のフロントまでの、ｚ軸に沿った距離を測定し、決定する（e_z）ことによって進行する。次いで、この距離と同程度の多くのｚスライスからなる第１の領域９２が創り出される。第１領域９２の後に続く各領域は、好ましくは、その前の領域の二倍深い。

【０１５９】

高品質スーパーサンプリングと組み合わせたとき、第１領域は、最終画像の１ピクセル当たり１つのレイを放射するとき、正確に、領域の終りで１ボクセル当たり１つのレイを持つのに必要ほど大きい。したがって、二倍より高いスーパーサンプリングが、高品質クローズアップ画面をレンダリングするのにボリュームのフロントに必要となるかも知れない。

【０１６０】

図１９に例示したように、視点１７２が、たとえば、ボリュームのフロント（すなわち、ｚ＝３領域境界）から３つのボクセル単位である場合、第１領域１７６は、好ましくは、３ボクセル単位厚となり、次の領域は、６ボクセル単位厚、以下同様となる。一般に、I番目の領域は、好ましくは、$e_z \cdot 2^{I}$ スライス厚である。ここで、e_zは、視点１７２からボリュームのフロント（図１９参照）までの距離である。領域をこのように定義することで、領域の任意のものを通してキャスティングされた任意２つの透視レイ１７４が、前部境界（すなわち、ｚ＝３境界）のところにあるときに二倍ほど後部境界（すなわち、ｚ＝２４境界）で離れ

ているという所望の効果を生み出す。これは、２つのレイ１７４間の距離が、第１領域１７６を横切って１単位から２単位まで成長し、次いで、４単位、最終的に、最終領域のところで８単位まで成長するように、図１９に示してある。さらに、領域境界１７０が大域的ジオメトリに依存しているので、このレイ・キャスティング・アルゴリズムの効率は、各ディメンジョンでボリューム解像度の一倍から二倍にレイ密度を保つための機構を設けることによって、最大となる。これは、また、透視レイがキャスティングされるときにデータのフィルタリングが制御されるようにも規則的なトポロジを創り出す。

【０１６１】

指数的距離のところに境界を持つ領域を設けることで、領域のバックのところよりもフロントで二倍高いレイ密度を生み出す。したがって、領域境界と交差するときにレイ密度を調整する機構を設けると好ましい。各レイが領域のリアでボクセル座標について出発すると好ましいので、領域のフロントでは、各ディメンジョンにおけるすべての第２レイは、好ましくは、ボクセル座標と直接一致する。残りのレイは、好ましくは、２つのボクセル位置間の途中で領域と交差する。この決定的なレイ・パターンを持つレイ密度をダウンサンプリングするために、二次元（２Ｄ）バートレット・フィルタ（テントまたは三角形フィルタとしても知られている）を、各ディメンジョンで±１ボクセルの範囲で、使用すると好ましい。各領域のフロントでのレイ密度がボクセル密度の二倍なので、この３×３ボクセル近傍は５×５レイと交差する。次に図２１を参照して、エッジ１７８の各々が、重みゼロであるから、３×３の隣接したレイ１８０のみを使用して、レイ密度をダウンサンプリングするのにフィルタを適用することができる。これは、隣接したレイを効果的に合併する。バートレット・フィルタは、最終的な画像で生み出す付加的な品質のための簡単なボックス・フィルタよりも好ましい。前から後への処理の場合は、レイは合併するかわりに分割される。ここで、レイのバイリニア補間が実施されて、他のレイ間で開始する新しいレイを生成する。ここで、±１のサイズのバートレット・フィルタが、バイリニア補間作業の逆のものであることは言っておかなければならない。

【０１６２】

図２２は、視点１９６がボリュームのフロントで（すなわち、ベースプレーン１９８から）３ボクセル単位であるとき、サイト・レイ１８６が7^3個のボリューム１９２を通ってどのように移動するかの２Ｄ例を示している。サンプリング・レートが１スライス当たり７と１４の間に留まっており、そして、レイ１８６が領域を通って後から前に移動するにつれて増大するということに注意されたい。N^3ボリュームについてレイ密度再サンプリング・ステージの数は、それがN^3ボリュームにおける最大領域数であるから、ので、のためのステージを再サンプリングしているレイ密度の数は、$\log_2 N$によって制限される。最終的な画像ワープが生じたとき、ベースプレーン１９８に示される最終再サンプリング・ステップが実施されると好ましい。

【０１６３】

図２２に示すように、ボリューム・データセット１８２のリアは、必ずしも常に領域境界１８４と一致するわけではない。しかしながら、レイ１８６が領域境界１８４のすべてで正確なボクセル座標１８８上に存在することが好ましいので、レイ１８６は、ボリューム・データセット１９２を囲む最終領域（陰影付き領域）のリアにおいてグリッド座標１９０上で発祥するのが好ましい。したがって、ボクセル座標およびレイ・サンプル場所１９４は、ボリューム１８２のリアに適合してはならない。これは、前述の境界条件を提供するだけでなく、視点１９６がボクセル単位距離より小さく動かされるときに一時的なアンチエイリアシングを助けもする。それは、レイ１８６がボクセルに対して同じ位置から発祥し続けるからである。

【０１６４】

図２３は、本発明の一形態に従って、ボリュームの後から前へのＥＲ透視投影を実施する好ましい方法を示しているが、このＥＲ透視方法の他の実施例も考えられる。上記のように、まず、目または視点からベースプレーンまでの距離を、好ましくは、決定する（ボクセル単位）。この視点位置を使用して、指数的領域境界を創り出す。次に、好ましくは、充分な領域を確立し、ボリューム・データセットを完全に取り囲む。ボリューム・レンダリングを実施するために、アルゴリズムは、後から前へ各領域を通してループし、ノーマル・レイ・キャスティン

グを計算するが、スライス・オーダー様式では、合成バッファに部分的に計算されたレイを記憶する。領域間（すなわち、領域境界）では、上記したように、合成バッファのレイ密度再サンプリングが実施されると好ましい。次に、ベースプレーン画像を、ディスプレイのために最終的な画像プレーン上へワープさせる。

【０１６５】

従来技術において知られている適応レイ密度透視方法の場合、一般的に、レイが不規則なパターンを使用して合併されるときに行われるフィルタ機能を決定することは困難である。しかしながら、本発明のＥＲ透視方法は、好ましくは、フィルタ動作のための規則的な境界を使用し、これらの境界内に正確にレイを配置するので、局所的バートレット・フィルタのカスケーディングによって行われる効果的フィルタを計算することはより容易である。これは、本発明のＥＲ透視アルゴリズムの重要な利点である。さらに、本発明の境界およびフィルタは、好ましくは、個別のデータについての普通の連続したフィルタリングと通常関連する低画質を克服するように選ばれている。

【０１６６】

たとえば、ここで、図２４に示すように、ボリュームの前方２ボクセル単位にある視点と共に７スライス分深いボリュームの透視投影を考える。本発明のER-透視方法を使用することで、領域を通してキャスティングされたレイ２００は、領域の後部のところで１ボクセル単位離れる。しかしながら、レイが領域境界２０２に達したとき、これらのレイは、好ましくは、局所的バートレット・フィルタを使用して濾過される。バートレット・フィルタ（１次に単純化された）は、サイズ2n+1のカーネルについて以下の重みを含み、出力が入力と同じスカラー範囲を有するように標準化される。

【０１６７】

【数３０】

$$0,\ \frac{1}{n^2},\ \frac{2}{n^2},\ \ldots,\ \frac{n-1}{n^2},\ \frac{n}{n^2},\ \frac{n-1}{n^2},\ \ldots,\ \frac{2}{n^2},\ \frac{1}{n^2},\ 0$$

領域境界での二次元画像の場合、本発明は、好ましくは、２つの主要方向に２つの一次元バートレット・フィルタを巻き込むことによって、二次元バートレット・フィルタを使用する。ＥＲ透視アルゴリズムは、好ましくは、常にレイを再サンプリングしてオリジナル密度の半分を有する。サイズ±２のフィルタを使用することで、レイ（n=2）は、５×５のフィルタ・カーネルまたは一ディメンジョンについて以下のたった５つ重みを創り出す。

【０１６８】

【数３１】

$$0,\ \frac{1}{4},\ \frac{2}{4},\ \frac{1}{4},\ 0$$

たとえば、図２４に示すように、サンプルａ、ｂ、ｃ、ｄ、ｅの、場所ｏで領域２から領域１へ変化する部分合成レイに対する寄与率を考える。

【０１６９】

【数３２】

$$o = \frac{1}{4}b + \frac{2}{4}c + \frac{1}{4}d$$

同様に、場所pおよびqでの部分的なレイを次のように計算する。

【０１７０】

【数３３】

$$p = \frac{1}{4}d + \frac{2}{4}e + \frac{1}{4}f$$

$$q = \frac{1}{4}f + \frac{2}{4}g + \frac{1}{4}h$$

ピクセルＡについての最終フィルタにおける重みがゼロなので、部分レイｎ、ｒ

についての式は省略されている。ＥＲ透視アルゴリズムを継続することで、再サンプリングされた部分レイｎ、ｏ、ｐ、ｑ、ｒは、好ましくは、領域１を通じてキャスティングされ、そこにおいて、局所的バートレット・フィルタによって再度濾過される。ｎ、ｏ、ｐ、ｑ、ｒの、ピクセルＡに対する標準化寄与率は、

【０１７１】

【数３４】

$$A = \frac{1}{4}o + \frac{2}{4}p + \frac{1}{4}q$$

ｏ、ｐ、ｑの値を代入して、

【０１７２】

【数３５】

$$A = \frac{1}{16}b + \frac{2}{16}c + \frac{3}{16}d + \frac{4}{16}e + \frac{3}{16}f + \frac{2}{16}g + \frac{1}{16}h$$

ここで、この式が、９つの値のカーネル・サイズ（ｎ=４）を持つバートレット・フィルタと同じ重み（すなわち、係数）を含むことは了解されたい。これは、同じフィルタ重みを持つピクセルＢについて繰り返すことができる。前から後への処理の場合、同様の分析を利用して、アルゴリズムの性能およびバイリニア補間の連続適用の結果を示すことができる。

【０１７３】

スライスの各サンプルは、好ましくは、同じ領域にある他の任意のサンプルと同じ量、最終画像に寄与する（カラー・マッピングおよび合成のようなサンプルに対する他のすべての動作が等しいと仮定する）。たとえば、サンプルｅの値は、局所的バートレット・フィルタのカスケーディング後の１／４の効果的な重みをもってピクセルＡに寄与する。同様に、サンプルＩは、１／４の有効重みでピクセルＢに寄与する。サンプルｆは、全体で１／４について３／１６の重みでピクセルＡに寄与し、そして、１／１６の重みでピクセルＢに寄与する。これは、

サンプルｇおよびｈについて繰り返すことができる。それぞれ、サンプルｅの左、サンプルＩの右のサンプルは、それぞれ、ピクセルＡの左のピクセル、ピクセルＢの右のピクセルに部分的に寄与し、最終画像に対するこれらの寄与率の合計もまた１／４となる。実際、この領域にあるすべてのサンプルは、同じ重みを有する。この領域がボリュームにおける第２の領域であるから、重みは１／４である。ボリュームの第１領域について、あらゆるサンプルは、好ましくは、１／２の重みを有する。これは、この領域の最終画像ピクセル当たり２つのレイがあることを実現することによって修正可能である。第２領域等における最終画像ピクセル当たりのレイは４つである。したがって、最終画像に向かう各ピクセルの寄与率を決定する重みは、（画像ピクセル）／（このスライス内のサンプル）となる。

【０１７４】

本発明のＥＲ透視方法がスライス・オーダー処理を実施するので、全計算量は、各スライスに実施される仕事量を算出することによって分析することができる。各サンプルについて行われた仕事が同じものであると仮定するならば、処理されるサンプルのカウント数を、作業負荷の比較として使うことができる。たとえば、オーバーサンプリング方法（図１８Ｂ参照）においては、領域境界上で正確に終わっているボリュームのリアにあるサンプル数は、N^2である。フロント・スライスでは、サンプルのカウント数は、視点のジオメトリに依存する。特に、類似した三角形を使用し、視点からボリュームのフロントまでの距離としてe_zと定義するならば、取り出されるサンプルの数は、

【０１７５】

【数３６】

$$\left(\frac{N^2 + N \cdot e_z}{e_z}\right)^2$$

これは、ボリューム・データセットを通して任意のスライスｓについて

【０１７６】

【数３７】

$$\left(\frac{N^2+N\cdot e_z}{e_z+s}\right)^2$$

に一般化することができる。したがって、オーバーサンプリング方法によって処理されるサンプルの総カウント数は、

【０１７７】

【数３８】

$$\sum_{s=0}^{N}\left(\frac{N^2+N\cdot e_z}{e_z+s}\right)^2$$

となる。同様に、アンダーサンプリング方法（図１８A参照）は、以下の仕事量を実施するように示すことができる。

【０１７８】

【数３９】

$$\sum_{s=0}^{N}\left(\frac{N\cdot e_z}{e_z+s}\right)^2$$

本発明のＥＲ透視アルゴリズムの場合、分析はより複雑となる。視野ジオメトリに依存して、

【０１７９】

【数４０】

$$\log\left(\frac{N+e_z}{e_z}\right)-1$$

領域が創り出される。先に示したように、これらの領域の各々は、好ましくは、$e_z \cdot 2^{-i}$スライスを含む。ここで再び、類似三角形のジオメトリック原理を使用するならば、本発明のＥＲ透視アルゴリズムは、以下の数のサンプルを処理する。

【０１８０】

【数４１】

$$\sum_{i=0}^{\log\left(\frac{N+e_z}{e_z}\right)-1} \sum_{s=0}^{e_z \cdot 2^{-i}} \left(\frac{N * (e_z * 2^{-i} - e_z + s)}{e_z * 2^{-i} - e_z} \right)^2$$

この式は、以下の上限を有する。

【０１８１】

【数４２】

$$\sum_{s=0}^{N} (2N)^2$$

そして、以下の下限を有する。

【０１８２】

【数４３】

$$\sum_{s=0}^{N} N^2$$

オーバーサンプリング方法（上記のもの）によって処理されるサンプルの総カウント数についての式を点検すると、視点がボリュームに非常に近いとき、オーバーサンプリング方法がフロント・スライスにＯ（N^2）作業を実行することがわかる。視点がボリュームのフロントにより接近するように移動するにつれて、

オーバーサンプリング・ランタイムが急速に成長する。上記のアンダーサンプリング式を調べると、視点がボリュームのフロントに接近するにつれて、分子がゼロに接近することがわかる。リア・スライスに実施される仕事量もゼロに接近する。アンダーサンプリング方法のランタイムは、視点がボリュームにより近くなるにつれて、減少する。

【０１８３】

視点ジオメトリに関係なく、本発明のＥＲ透視アルゴリズムによって実施される仕事量は、１スライス当たりｏ（N^2）およびω（$4N^2$）だけ制限される。この方法のいくつかの利点は、アルゴリズムのランタイムについての上限がボクセルの数についてリニアであり、視野位置から独立しており、また、達成される画質についての下限も視野位置とは無関係であるということである。したがって、ユーザは、所望の画質について基本サンプリング・レートを設定することができ、この所望の画質についてのボリュームの全体にわたってサンプリング・レートが充分であることが確実になる。

【０１８４】

それと対照的に、普通のオーバーサンプリング方法は、画質についての下限を提供するが、アルゴリズムのランタイムは、本発明のＥＲ透視方法のランタイムよりもかなり大きくなり得る。普通のアンダーサンプリング方法は、レンダリングのためのランタイムについての上限を提供するが、画質はＥＲ透視方法よりもかなり悪くなる可能性がある。

【０１８５】

図２３を再び参照して、ここには、本発明による、好ましい後から前のＥＲ透視なレイ・キャスティング・アルゴリズムが示してある。図２３のアルゴリズムは、擬似コード表現として示され、Ｚ主要軸線投影を採る。本発明のＥＲ透視アルゴリズムは、均一で規則的なグリッド上へ透視投影を実施するときに伝統的な落し穴に苦しむことがない。このユニークな方法は、オーバーサンプリング方法よりも急速に実行でき、アンダーサンプリング方法よりも良好な画質を与える。レイを合併するためのバートレット・フィルタを使用することで、普通ボックス・フィルタよりも画質を向上させることができる。ＥＲ透視アルゴリズムは、入

力データについての効果的フィルタリングを特徴づけることによって適格となる。

【０１８６】

本発明の別の形態によれば、大きなボリュームをレンダリングする方法が提供される。この方法では、ボリューム・データセットが本発明のＣｕｂｅ－５装置の物理的なシングル・パス能力を超える。この好ましい方法は、ボリューム・データセットを複数の立方形レンガに細分する。所定のオーダーでレンガを移動させることは、好ましくは、レンダリング前に先のレンガのベースプレーン画像でＣｕｂｅ－５装置の合成バッファの初期化を可能にする。それによって、レイ・パスおよび合成は、ボリューム全体を通じてロジック的に拡張される。レンガ間の境界に関する情報は、好ましくは、正しいサンプリングを保証するために再読み込みされる。この方法を使用して、最大ボリューム・サイズは、利用できる中間ベースプレーン記憶量によってのみ制限される。

【０１８７】

透視投影中に複数のボクセルが同じ画像ピクセルに寄与するデータセットの領域においては、同等の品質の画像が、好ましくは、ディテール・レベル（ＬＯＤ）ツリーを使用してレンダリングされる。このツリーは、たとえば、事前処理段階で増大する近傍サイズのボクセルを結合することによって生成してもよい。ＬＯＤを利用して単一の大きいボリュームを透視レンダリングしている間、好ましくは、視点にかなり近い、ボリュームの小さい部分だけが、最高精度で読み込まれなければならない。視点に関して、ボリュームのより遠い部分を、次に、データのより低い解像度バージョンからレンダリングされ得る。したがって、フレーム率および／またはデータセット・サイズは、好ましくは、増大する。本発明の透視アルゴリズムにおける各領域（先に述べた）は、今や異なったＬＯＤのところにあるので、領域間でレイを濾過する必要はもはやなく、レイを再分布させることだけが必要である。好ましくは、各ＬＯＤツリー・レベルの１つの領域だけが処理され、したがって、これらの領域だけをメモリ内にページングしなければならない。

【０１８８】

本発明のディテール・レベル（ＬＯＤ）方法は、また、、視点から異なった距離で複数のオブジェクトからなるシーンをレンダリングするのにも使用できる。このような場合、開始ＬＯＤは、好ましくは、スクリーン・スペース画像とほぼ同じサイズのベースプレーン画像をデリバーし、レンダリング時間を画像解像度に関連付け、オブジェクト・サイズに関連付けない（すなわち、スケール独立性）ように選ぶ。

【０１８９】

後から前へのレンダリングも同様に本発明の意図したものであり、本発明の範囲内に入っているが、ユニークなＬＯＤ方法は、ここでは、前から後へのレンダリング・コンテキストとして説明する。前から後にレンダリングすることで、レンダリングしようとしているボリュームの最も詳細な表現のスラブで週発することが好ましい。本発明の好ましい方法においては、ボリューム・スラブの厚さは、スラブのフロントおよびバックにおける投影ボクセル距離が、先に説明したような本発明の透視投影と同様に、２の因数だけ異なるように選ぶ。スラブをレンダリングした後、現合成バッファ画像をワープ単位で０．５の因数だけスケーリングすると好ましい。これは、半解像度の次のスラブのレンダリングのために合成バッファを初期化する。好ましくは、各ＬＯＤのスラブは、１つだけ、実際にレンダリング・パイプラインを通って流れる。したがって、大きいボリュームの場合、これらのスラブだけが、オンボード３Ｄメモリ内にページングされなければならない。

【０１９０】

ここで、本発明の装置がオフライン計算（たとえば、ディテール・レベル（ＬＯＤ）の生成やデータセットのフィルタリング）をスピードアップするのにも使用できることは明らかである。ＬＯＤを生成するためには、本発明のトリリニア補間ユニット（Trilin）は、好ましくは、そのすべての重みを０．５にセットする。ひとたび新しいサンプルを利用できるようになったならば、これらのサンプルを、好ましくは、サブサンプリングし、新しいボリュームにコンパクト化する。これが、次のより粗いＬＯＤとなる。データセットを濾過するために、トリリニア補間ユニットは、再び、０．５の重みだけを使用する。しかしながら、この

とき、データは圧縮なしにレンダリング・パイプラインの始めに戻される。各付加的なパスは、すべての主要軸線方向にもう１つのボクセル範囲を有するフィルタ・カーネルを備える新しい濾過済みのボリュームを創り出す。

【０１９１】

より高い品質の画像をレンダリングするために、本発明の装置および方法は、好ましくは、所望のトレードオフに依存して、フル・ハードウェア手段、マルチパス・アルゴリズムおよび／またはこれら２つの組み合わせを利用する融通性を提供する。フル・ハードウェア手段およびマルチパス方法は、好ましくは、２つの主要機能領域、すなわち、フィルタリングと補間におけるより正確な計算を提供する。

【０１９２】

Ｃｕｂｅ－４アーキテクチャ、すなわち、本発明（Ｃｕｂｅ－５）の先行モデルは、或る特定の場所でのｘ、ｙ、ｚ勾配の各々を評価するために２つだけのサンプル・ポイントを持つ中央差分勾配フィルタを利用する。より大きい３Ｄフィルタが、より正確な勾配評価をデリバーすることができる。たとえば、Ｓｏｂｅｌフィルタ（中心ポイントからのマンハッタン距離の逆数から誘導した重みを持つ3^3フィルタ）である。しかしながら、3^3フィルタの直接的なハードウェア機能は、２７台の乗算器と２６台の加算器を必要とする。

【０１９３】

本発明の装置は、対称形回旋フィルタを使用することによってこの高価な従来技術方法の代替案を提供する。この回旋フィルタは、３台の乗算器と６台の加算器のみを能率的に備えることが可能であり、これはコスト節減にとって意味があることである。１勾配成分当たりのハードウェアの複製は、好ましくは、代わりに３パス・アルゴリズムを適用することによって避けることができる。たとえば、図２５は、Ｓｏｂｅｌ勾配フィルタのｘ成分の対称形近似を示している。各ステージ内で、重みは、好ましくは、合計前に最も近い隣接体に適用される。図２５を参照して、各ステージが生データの代わりに前のステージの出力に作動する場合、図２５に示した重みは、Ｓｏｂｅｌ勾配フィルタの3^3対称形近似（図２５の右側）を効果的に生成する。ｘ重みを｛１ｗ１｝に変えることで、その代わ

りに、ガウス・フィルタの近似を生成することになる。

【０１９４】

本発明は、付加的なハードウェアを必要としないが、フレーム率を低下させるより高い品質のレンダリング・モードを意図している。このような例の１つは、ディテール・レベル（ＬＯＤ）情報を利用することによって勾配評価へのより大きい近傍寄与率を達成することである。中央差分勾配が次のより粗いＬＯＤのデータについて計算される場合、６×４×２フィルタを同等に効果的に使用する。この場合、６は、現勾配成分の方向における範囲にある。本発明の装置（すなわち、Ｃｕｂｅ－５アーキテクチャ）は、データのミップ・マッピングしたＬＯＤ表現を保持することができるので、このフィルタは、単純な中央差分解を越えて、本質的にハードウェアの増加なしに、達成することができる。

【０１９５】

本発明によって提供される別のより高い品質のマルチパス・レンダリング・モード（付加的なハードウェアは必要ない）は、医療分野やその他の分野で有利な用途を有するトリキュービック補間の近似である。このモードは、より正確な再サンプリングおよびイソポジション計算を可能にする。これのために、本発明は、好ましくは、ピースワイズ4^3ボクセル・フィルタを一連のリニア補間および外挿に分解する。これは、すべてのディメンジョンで対称的であり、中間結果の効果的な再利用を可能にする。

【０１９６】

より高い品質のレンダリングを実施する際して、多パス・アルゴリズムを使用するのとは逆に、Ｃｕｂｅ－５パイプライン内により正確で融通性のある勾配評価を与えるための付加的なハードウェアを使用することには、或る種のトレードオフがあることは了解されたい。一般に、多パス・アルゴリズムを使用することで、本発明のアドレス生成・制御ユニット（図５参照）を変えて計算目的のためのパイプラインを一時的に停止させることが必要であり、それと共に、ハードウェア方法が付加的な特定用途向け集積回路（ＡＳＩＣ）ロジックと、より大きい近傍をサポートするための付加的なコネクションとが必要である。

【０１９７】

強化したボリューム・レンダリング能力に関して、本発明の好ましい実施例は、任意のプレーンによるクリッピングをサポートする。各プレーンからの距離は、好ましくは、１プレーン当たりレジスタと１つの加算器だけを使用して増分計算することができる。目で見えるように正方向だけを定める普通のクリッピング・プレーンに加えて、本発明の装置は、好ましくは、所定オフセットの外側に位置するすべてのサンプルを無効にすることによって、斜めのマルチ平面再フォーマット化（ＭＰＲ）のためのデータセットから任意の厚いスライスを抽出することをサポートする。

【０１９８】

軸線整合したカッティング・プレーンが、当該立方体に対して横方向にボリュームを制限することによって、実施されると好ましい。あるいは、本発明は、ボリュームからの単純な立方体を排除するようにこの横断を制限すること（たとえば、ボリュームのほとんどの八分円を視覚化すること）も意図している。

【０１９９】

クリッピングに加えて、本発明は、さらに、オブジェクトのカラーを変調して、たとえば、半透明の媒体を通る光の大気減衰をシミュレーションするデプス・キューイングを意図している。当業者によって明らかなように、この現象は、媒体も或る種のカラー（たとえば、白色または灰色）に寄与するとき、霧またはかすみと呼ばれる。本発明に従ってこの特徴を実施するために、通常、転送機能を修正することによって、はっきりした領域を半透明なカラーと共に置くと好ましい（たとえば、デプス・キューイングの場合には黒、霧の場合には白）。各最終ピクセルは、好ましくは、さらに、減衰され、視点からボリュームの表面までの距離を勘案する。これは、ワーピングの一部として実施されると好ましい。

【０２００】

本発明の装置は、さらに、ｘ、ｙ方向における２の好ましいデフォルト・スーパーサンプリング・レートでスーパーサンプリングした画像のレンダリングをサポートするが、他のサンプリング・レートも考えられる。画質をさらに向上させるために、各レイに沿ったサンプリング・レートを増大させてもよい。どちらの方法も、３Ｄメモリからボクセルを再読み込みする必要がない。本発明の装置は

、好ましくは、ボリューム横断オーダーを変え、既にバッファ内に存在するボクセルを繰り返し読み出すようにする。ボクセルを再利用されるたびに、新しい重みが、好ましくは、新しい再サンプリング位置を反映するように本発明のトリリニア補間ユニット（Trilin）で利用される。

【０２０１】

本発明における好ましいスーパーサンプリング実施例においては、中央差分勾配が、１距離単位隔たった隣接体間で計算され、充分な精度を保証する。これらの勾配は、好ましくは、最初に差分を取り出し、その後補間することによるか、または、最初に補間し、次いで、ｋ個の位置隔たった（ｋ×オーバーサンプリングと仮定する）隣接体間の差分、好ましくは中間隣接体のではない差分を採用することによって計算される。新しいα′値を計算するとき、分類ステージは、新しいサンプル間距離を考慮しなければならない。したがって、スーパーサンプリング中、ボリュームは、好ましくは、各スライス内のインタリーブされたパターンにおいて横断させられることになる。これは、本質的に、半透明の材料（ゲル）がサンプリング・レートから独立して蓄積された不透明度（ＲＧＢα値）を保つことを保証する。したがって、たとえば、ｚ方向におけるkのオーバーサンプリング因数の場合、修正済みのα′値が使用されると好ましい。ここで、$\alpha' = 1-(1-\alpha)^{1/k}$。

【０２０２】

異方性データセットは、異なった軸線に沿ったサンプル間の異なった距離を有する。したがって、勾配計算および最終的な二次元（２Ｄ）画像ワープは、好ましくは、軸線依存スケーリング因数を必要とする。それに加えて、サイト・レイがボリューム・データセットを通してキャスティングされつつある方向は、好ましくは、暗黙のボリューム・スケーリングを勘案するように調整を必要とする。これは、等方性グリッドに異方性データを記憶するときに生じる。α′値は、好ましくは、サイト・レイがボクセル・セルを通して移動する方向依存距離ｄに従って、調整される。訂正されたα′は、$\alpha' = 1-(1-\alpha)^{d}$である。ここで、方向依存距離ｄは、好ましくは、範囲

【０２０３】

【数４４】

$$[1, \sqrt{3}]$$

内にある。

【０２０４】

上記のボリューム・レンダリング能力を強化する方法に加えて、本発明は、さらに、ポリゴンとボリュームの混合、ポリゴンのボクセル化、複数のオーバーラップしているボリュームのレンダリング、テクスチャ・マッピングの実施および画像ベース・レンダリングの加速を含むユニバーサル三次元（３Ｄ）レンダリングを行ういくつかのユニークな方法を提供する。これらの方法を、より詳しく以下に説明する。

【０２０５】

本発明の重要な局面は、単一の画像においてジオメトリック・オブジェクト（すなわち、ポリゴン）およびボリュームを正しく混合するそのユニークな能力にある。本発明の装置（すなわち、Ｃｕｂｅ－５）は、好ましくは、普通のジオメトリ・ハードウェアに影響を及ぼして、Ｃｕｂｅ－５ボリューム・レンダリング・パイプラインと共に不透明、半透明のポリゴンをレンダリングする。

【０２０６】

本発明の好ましい方法においては、ボリュームおよび不透明なポリゴンを含むシーンをレンダリングするために、すべての不透明なポリゴンを、まず、Ｚバッファ上に投影する。このＺバッファは、所定のベースプレーンと一致し、ボリューム・サンプル距離と一致するに充分な解像度を有する。Ｚバッファを使用することで、好ましくは、ボリュームのどのスライスがベースプレーン画像の各ピクセルについてポリゴンのフロントにあるかついて判定がなされる。次に、好ましくは、合成バッファをプリロードする（すなわち、初期化する）。このとき、この投影されたＲＧＢαＺ（すなわち、Ｚバッファ）画像は、ポリゴンのカラーおよびデプス画像を表している。その後、ボリュームを合成バッファにおいて可能にされたｚ比較でレンダリングする。不透明なポリゴンのデプス値をチェックし

て、不透明なポリゴンで隠されたボリューム・サンプルを最終画像に寄与させ続ける。最終的に、不透明なポリゴンは、背後にボリュームを隠し、フロントにあるボリュームがポリゴン上に正しく合成される。

【０２０７】

換言すれば、本発明の好ましい方法によれば、合成バッファはｚバッファ画像{C_z、Z_z}でプリロードされる。ここで、C_zは、ジオメトリ・サンプルの値を表しており、Z_zは、所定視点からジオメトリ・サンプルのデプスを表している。後から前への合成中、ボリューム・サンプルがジオメトリの後にあるとき（すなわち、サンプルZ_sのデプスがジオメトリ・デプスZ_zより大きいとき）、合成バッファC_{out}における生じた出力ピクセルは、好ましくは、ジオメトリ・サンプル値C_zに等しくなる。同様に、前から後への合成中、当業者には明らかなように、サンプルは、好ましくは、Porter-Duff over operatorを使用して合成される。Porter-Duffα合成ルールについての詳しい説明は、たとえば、テキストCompositing Digital Images, by T. Porter and T. Duff, Computer Graphics (SIGGRAPH84), vol. 18, no. 3, pp. 253-259, July 1984にある。この文献は、参考資料としてここに援用する。したがって、合成バッファの結果生じた出力ピクセルC_{out}は、好ましくは、ボリューム・サンプルがジオメトリのフロントにあるとき（すなわち、ボリューム・サンプルZ_sのデプスがジオメトリ・デプスZ_zより小さいとき）、ジオメトリ・サンプル値C_sを通じてボリューム・サンプル値C_sに等しくなる。

【０２０８】

すべての断片（半透明のポリゴン・ピクセルおよびボリューム・サンプルの両方）は位相的デプス・ソーテッド・オーダーで描かれなければならないので、半透明ポリゴンは、より複雑な状況を採る。これは、over operatorを持つ合成半透明断片は交換可能でないので、必要である。したがって、ポリゴンは、シーンまたは視野ジオメトリが変わるときはいつでも、リデプス・ソートされなければならない。さらに、ソーティングは、デプス・サイクルの取り扱いを含めて、位相的に正しくなければならない。

【０２０９】

或る種のハードウェア・ソーティング・サポートを与えるためにＡバッファを使用するアーキテクチャが提案されているが、ハードウェアにおいてＡバッファを実装することで、一回だけのパスでほんの限られたデプス複雑性（すなわち、１ピクセル当たりのオーバーラップしているポリゴンの数）を可能にするが、これは高価である。たとえば、普通のＡバッファ・アルゴリズムの説明は、たとえば、テキストThe A-Buffer, an Antialiased Hidden Surface Method, by L. Carpenter, Computer Graphics （SIG GRAPH 84）, vol. 18, no. 3, pages 103-108, July 1984に見出すことができる。

【０２１０】

好ましい方法において、本発明は、スライス・オーダー・レイ・キャスティングにポリゴン・レンダリングを適応させ、ポリゴン毎あるいはピクセル毎よりもむしろ、ボリューム・スライス毎に全レンダリング・プロセスを同期化する。Ｃｕｂｅ－５装置は、好ましくは、図２６に示すように、ボリューム・サンプル２１２のスライス間にインタリーブされた、あるいは、ぴったり適合する薄いスラブのジオメトリック・オブジェクトをレンダリングするのにジオメトリ・パイプラインおよび普通のグラフィックス・ハードウェアを利用する。

【０２１１】

次いで図２６を参照して、ボリュームの各スライスは、好ましくは、ボリューム保存軸線に対して垂直なプレーンでサンプリングされる。これらのプレーンは、目または視点２１４から最も遠いところから目に最も近いところまでデプス・オーダーで（たとえば、遠近クリッピング・プレーンを使用することで）描写される。したがって、ボリュメトリック・データと半透明のポリゴンを混合するために、ポリゴン２１０の薄いスラブは、好ましくは、ボリューム・サンプル２１２のスライス間にレンダリングされ、合成される。ここで、スラブ２１０がボリューム・サンプル・プレーンの２つの連続したスライス間に位置する半透明オブジェクトのすべてを表すことは明らかである。スラブの境界は、好ましくは、すべてのレンダリングされたスラブ２１０の結合が任意の領域（たとえば、図２６に示すような、〔〕，〔〕，　〔〕）を失ったり、重複させたりすることがないように定められる。ボリューム・スライスおよび半透明ポリゴン・スラブ２１０からの

データは、交互に相互にぴったりと適合させられる。こうして、すべての寄与している実体の正しいデプス・オーダリングが保存され、それらを合成するのにover operatorを使用して最終画像ピクセルに正しいカラーを創り出す。

【０２１２】

本発明の好ましい方法によれば、不透明なポリゴンは、まず、Ｚバッファリングで描かれる。任意のボリューム・スライスを描く前に、ボリューム範囲の後に位置する半透明のポリゴンが、好ましくは、任意普通の半透明ポリゴン・レンダリング・アルゴリズム（たとえば、ペインタ）を使用して不透明ポリゴン上に描かれる。同様に、ボリュームのフロントに位置する半透明ポリゴンは、アルゴリズムの混合部分の後に描かれると好ましい。ボリューム境界内にデプス方向に位置するが、ボリュームのトップ／ボトム／サイドまでデプス方向に位置するポリゴンは、好ましくは、ボリューム・スライスが半透明ポリゴンを無限にカッティングするように延びるプレーンであったかのように、スライス・オーダーで描かれる。

【０２１３】

OpenGLを使用して、半透明ポリゴン・オブジェクトの薄いスラブを直接的にレンダリングしてもよい。ポリゴンは、OpenGLに含まれるグーロー・シェーディング模型を使用してシェーディングされると好ましい。素朴な方法としては、すべてのスラブについて半透明ポリゴンの完全セットをレンダリングし、データの現行の薄いスラブを切るようにより高くてもっと遠くのクリッピング・プレーンをセットすることである。しかしながら、n^3ボリュームの場合、レンダリングしなければならないｎ個までの薄いスラブがあり得る。代表的なシーンが薄いスラブのすべてにわたる極めて少ないポリゴンを含んでいるので、本発明は、ポリゴンをスラブ境界にクリッピングし、各スラブ内のポリゴン部分をレンダリングするだけの代替方法も意図している。これは、実質的に、ポリゴン・パイプラインにかかる処理負荷を減らす。しかしながら、各薄いスラブの２つのプレーンに対してすべてのポリゴンをクリッピングするアプリケーションが必要となる。

【０２１４】

図２７に示すように、本発明は、２つのクリッピング・プレーン２１６、２１

８が平行であって、プレーン間に位置するポリゴン部分のみを保持するという事実の利点を採用することを意図している。これは各プレーンに対して別個にクリッピングするよりも少ないポリゴンを創り出すけれども、なお、劇的に三角形カウントを増やすことができる。第１のケースは、三角形２２０が薄いスラブと交差するが、頂点はスラブ境界２１６、２１８内にはまったくないときに生じる。これが生じたとき、１つの頂点は、スラブの片側にあり、他の２つ頂点は、スラブの反対側にあり、２つの三角形に分解される台形を創り出さなければならない。次に、三角形の１つの頂点がスラブ内にあるときを考える。１つの状況において、残りの２の頂点が現スラブの同じ側に位置するように、三角形２２２はスラブと交差し、たった１つの三角形を作る。第２の状況において、残りの２つの頂点が現スラブの反対側に位置するように、三角形２２４がスラブと交差する。これは、五角形または３つの三角形を生じさせるので、最悪のケース状況である。最終ケースは、２つの頂点が同じスラブ内に位置するように三角形２２６がスラブと交差するときに生じ、ここでも再び、台形が創り出され、２つの三角形を生じさせることになる。

【０２１５】

本発明の好ましい実施例において、半透明ポリゴンにバケット・ソーティング方法が適用される。視野ジオメトリが変わるときはいつでも、ボリューム・サンプルの配置がそれらのジオメトリに対する位置を変える。したがって、本発明は、好ましくは、２つのボリューム・サンプル・プレーン間に各薄いスラブについてバケットを創り出す。シーンにおける半透明ポリゴンのすべては、好ましくは、横断され、各ポリゴンが、それの交差するスラブの各々についてバケット内に置かれる。たとえば、図２８に示すように、三角形Ｔ１は、全部で６つのスラブＳ１～Ｓ６をスパンしているので、全部で６つのバケット内に置かれる。三角形Ｔ２は、スラブＳ２、Ｓ３に対応するバケット内に置かれ、残りの三角形についても同様に行われる。図２８に示す例の場合、４つの三角形Ｔ１～Ｔ４をバケットに入れることで、グラフィックス・パイプラインに送られる三角形が１２個となる。それと比較して、三角形がスラブ境界にクリッピングされている場合、２０個の三角形がグラフィックス・パイプラインに送られることになろう。

【０２１６】

バケット法に代わる案は、走査変換ポリゴンで利用される能動エッジ・リストと同様の能動三角形リストを作ることである。三角形は、それらが交差する第１のスライスのところで能動リスト内に置き、三角形がもはや任意のスライスと交差しなくなったときにリストから外すことができる。各三角形が最初にどのスライスに遭遇したかを示すデータ構造を事前計算すると好ましい。この前処理は、本質的に、バケット法と同じであるが、バケット法では、各スライスについて三角形除去をチェックしなくてもよい。

【０２１７】

本発明の方法の１つ利点は、所定フレーム率を維持するために画質をトレードオフする方を選ぶ用途の場合、ボリュームについて描かれるスライスの数が減少するにつれて描かれるポリゴンの数が減少するということである。ボリューム・スライスの数が減少するにつれてインタースライス・サイズが増加するため、これが生じる。達成されるレンダリング率は、描かれるポリゴンの数および描かれるボリューム・サンプルの数（これは、描かれるボリューム・スライスの数に比例する）にほぼ比例する。このトレードオフから生じる画質劣化は、任意のボリューム・レンダリング・アルゴリズムにおいてより少しのサンプルを取り出すことと同様に、ボリューム・データにのみ影響を及ぼす。

【０２１８】

半透明のジオメトリおよびボリュームを混合するとき、１つの薄いスラブ内の同じピクセルに描かれつつある２つまたはそれ以上の半透明ポリゴンを取り扱うためオプションが少なくとも３つ存在する。第１のオプションにおいて、ポリゴンは、over operatorをもって規則的な処理オーダーで描くことができる。この方法は不正確なカラーを生成する可能性があるが、ポリゴンが薄いスラブ内にバケッティングすることによってなおソートされるので、カラー・エラー量は制限される。

【０２１９】

２つまたはそれ以上の半透明ポリゴンを取り扱う別の方法は、オンザフライ・ボクセル化として２つのボリューム・サンプル・スライス間に半透明ポリゴンの

薄いスラブを描くことである。普通のボクセル化方法では、表面が３Ｄボリューム・グリッドに変換された３Ｄ走査であるとき、グリッドの解像度は、普通、単一のボクセルのサイズが、レンダリング時に人間の目によって識別され得る最も小さい面積を表すように、選ばれる。Ｘ、Ｙディメンジョンにおいて、ポリゴンは、スクリーン解像度まで描かれる。Ｚディメンジョンにおいては、各ボリューム・サンプルが充分なスライスをもってレンダリングされ、各ボリューム・サンプルもまた人間の目によって識別され得る最小面積を表すと仮定される。したがって、Ｚディメンジョンにおける２つのボリューム・スライスによって境される各ピクセルもこの小さい面積を表す。

【０２２０】

前述のことに鑑みて、本発明の一実施例に従って実施される方法は、３Ｄグラフィックス・ハードウェアを利用することによってオンザフライ・ボクセル化を計算するものと見ることができる。ボクセル化方法は、２つの好ましい方法のうちの一方を使用することによって、ポリゴンを単一のボクセルに結合する。第１の方法は、各カラー・チャネルのｍａｘを取り出すことである。第２の方法は、

【０２２１】

【数４５】

$$C_v = \frac{(C_{p1}D_{p1} + C_{p2}D_{p2})}{(D_{p1} + D_{p2})}$$

として重み付きｍａｘを取り出すことである。ここで、C_{pl} は、第１ポリゴン（ポリゴン１）のカラーであり、D_{pl} は、ポリゴン１の密度であり、C_vは、ボクセルに割り当てられたカラーである。たとえば、多くのOpenGL手段は、ｍａｘをglBlendEquationext（gl_max_ext）とブレンドするのを可能にする。密度がアルファ値（たとえば、ボリューム・レンダリングのためのリニア・ランプ転送機能）に等しいと仮定するならば、カラーは、好ましくは、glBlendFunc(gl_src_alpha, gl_one)を用いることによってブレンド前にそれらのアルファ値で重み付けすることができる。しかしながら、OpenGLは、蓄積後にアルファ値の合計によっ

て割ることができないので、完全な先の式を計算することもできない。

【０２２２】

１つの薄いスラブ内の同じピクセルに２またはそれ以上の半透明ポリゴンを描画する第３の方法は、最も正確な方法と考えることができる。デプス・ソーティング（たとえば、ＢＳＰツリー）を実施するために本発明の先に述べた方法の１つを利用することによって、各スラブ内のすべての半透明ポリゴンを適切にオーダーリングし続けることができる。デプス・サイクルは、好ましくは、区割りで使用されるプレーンをスパンするポリゴンを分割することによってＢＳＰアルゴリズムで取り扱われる。そして、結局、サイクル内のポリゴンの１つが区割りプレーンとして使用される。

【０２２３】

前述のように、本発明のＣｕｂｅ－５の重要な特徴は、少なくとも１つのジオメトリ・パイプラインまたはエンジンをＣｕｂｅ－５システムに接続するユニークな能力である。本発明によれば、ＰＣクラス・マシンについて請求項記載のＣｕｂｅ－５システムに１つまたはそれ以上のジオメトリ・パイプラインを接続する２つの好ましい方法が、以下に説明するように、提供される。両方法は、不透明および／または半透明ポリゴンをボリュメトリック・データと混合するユニークな方法である。

【０２２４】

ここで、不透明なポリゴンが、好ましくは、ボリューム・データセットを通して投影した後、ワーピングが最終画像上に正しいフットプリントを創り出すということは明らかであろう。さらに、Ｚデプス値は、好ましくは、処理軸線に沿って整合させられ、その結果、ボリューム・スライス・インデックスを・チェックのために使用できる。

【０２２５】

本発明の一実施例によれば、好ましい方法は、現視野方向について主要視野軸線を決定することによって開始する。図２９に示すように、主要視野軸線２３０が、たとえば、Ｚ軸に沿うように、好ましくは、変換がジオメトリ２２８に適用される。次に、視点またはアイ・ポイント２３２は、好ましくは、Ｘ軸まわりの

所定角度αとＹ軸まわりの角度βの分だけ、注視ポイント２３４とアイ・ポイント２３２との間でベクトルを回転させることによってこの方向に沿うように移動させられる。好ましくは、α、βは、常に、－４５～＋４５度の範囲にある。そうしないと、異なったベースプレーンが選ばれることになろう。ＸおよびＹに沿ったＺスライス・シャー変換（「Ｘに従うＸ、Ｙ」シャーとしても知られている）は、好ましくは、その後に、次のように視野マトリックスに適用される。

【０２２６】

【数４６】

$$\begin{bmatrix} 1 & 0 & \tan\alpha & 0 \\ 0 & 1 & \tan\beta & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

このジオメトリについて、不透明なポリゴンを描くとき、ポリゴン・フットプリントが「プレワープされる」。その結果、Ｃｕｂｅ－５レンダリングの終りでワーピング作業が最終的な画像に正しいポリゴンを創り出す。さらに、計算されるＺデプスは、好ましくは、処理軸線に沿って距離に比例する。（たとえば、すべての不透明ジオメトリがボリューム範囲内に適合するならば）、ボリュームのエッジにあちこちのクリッピング・プレーンをセットすることが可能となり、そして、デプス・バッファの精度が同じである場合、計算されたデプスは、デプス・チェックについて正確にボリューム・スライス・インデックスとなる。さもなければ、計算されたデプスがボリューム・レンダリング・システムによって利用されるとき、単純なスケーリングを適用しなければならない。しかしながら、この方法を使用するとき、シャーリングは光を正しい場所に移動させることができないので、光位置を考慮しなければならない。

【０２２７】

半透明ポリゴンの薄いスライスは、好ましくは、スペース内のそれらの３Ｄ位置とジオメトリック的に整合する。好ましくは、アイ・ポイントが、前述したよ

うに、最初に整合させられる。次に、オブジェクトが最終画像プレーンへ全面的に投影するのを防ぐために、ジオメトリは、好ましくは、現在の薄いスラブの中心がシャーリング前にＺ＝０となるように並進させられる。クリッピング・プレーンによって、現在の薄いスラブだけをレンダリングすることができ、投影プレーンは、たとえば、glOrtho（透視の場合、glFrustum）を持つ領域を境する２つのボリューム・スライス内にあるようにセットされる。

【０２２８】

本発明を理解するには、フレーム・バッファ設計およびコンポスティング・バッファ設計の構成を理解することが重要である。図３０に示すように、本発明のＣｕｂｅ－５ボリューム・レンダリング・パイプライン２３６は、好ましくは、コンポスティング・バッファと呼ばれる緊密に接続されたオンチップＳＤＲＡＭバッファ２３８を利用し、ボリュームがスライス・オーダーで処理されるときに部分的に合成されたレイを保持する。このアーキテクチャは、スライス・オーダー・レンダリングに固有の規則的な処理シーケンスを利用する。具体的には、ボリューム２４０の各スライスは、好ましくは、各行の先の最も左のボクセルから最も右のボクセル、そして、各スライスの最も下の行から最も上の行一番上の列（おそらく若干のスキューを持つ）と同じオーダーで処理される。こうして、ＳＤＲＡＭコンポスティング・バッファ２３８は、スライスのサイズに等しい長さを有する単純なＦＩＦＯ待ち行列になる。ＳＤＲＡＭ待ち行列は、好ましくは、３２ビット長であり、８ビットの固定ポイントＲＧＢα値（コクセルと呼ばれる）を保持することができる。各パイプライン２３６は、好ましくは、待ち行列のフロントからコクセルを読み込み、１つのコクセルをクロック・サイクル毎に待ち行列のリアに書き込む。

【０２２９】

それと対照的に、次に図３１を参照して、普通のＰＣクラス・ジオメトリ・パイプライン２４２は外部のＤＲＡＭフレーム・バッファ２４４を利用する。これは、各ピクセルについてＲＧＢαカラー値およびＺデプス値を記憶する。ポリゴン・レンダリングがスライス・オーダー・ボリューム・レンダリングに固有の規則的なアクセス・オーダーリングを好まないので、このフレーム・バッファ２４

４は、ランダム・アクセスをサポートしなければならない。ノーマル・ポリゴンレンダリングは、１０～５０個のピクセル間で平均のスクリーン上に三角形を生成する。したがって、ＤＲＡＭメモリは、このサイズのスクリーンの面積へのアクセスを最大にするように構成される。

【０２３０】

図３１に示すように、本発明の３Ｄテクスチャ・マッピング方法をジオメトリ・パイプライン２４２に実施したとき、スクリーンに対して垂直なボリューム・スライス２４６は、ボリュームを通してテクスチャ・マッピングされる。ボリューム・スライス２４６についてのペル頂点ジオメトリ計算は、任意レベルのグラフィックス・ハードウェアで容易に達成することができる。しかしながら、テクスチャ・メモリ２４８およびフレーム・バッファ２４４の両方に対するランダム・アクセスをサポートする要件は、ＤＲＡＭフレーム・バッファで達成できる塗り潰し率にこの方法の性能を限定する。

【０２３１】

非常に高い端面グラフィックス・システムは、代表的には、ポリゴン・パイプラインの断片処理セクション２５０における大規模な平行度を利用する。これは、非常に分散されたフレーム・バッファと相まって、塗り潰し率性能を向上させることができる。

【０２３２】

図３２には、本発明に従って、Ｃｕｂｅ－５ボリューム・レンダリング・システム２５２にジオメトリ・パイプライン２４２を接続するための一実施例が示してある。図３２に示すように、ＳＤＲＡＭコンポスティング・バッファは、好ましくは、Ｃｕｂｅ－５パイプライン２５２内部から除去され、その代わりに、外部のＤＲＡＭフレーム・バッファ２５４を使用する。普通のポリゴン・エンジンと同様にＤＲＡＭフレーム・バッファ２５４を構成するというよりもむしろ、本発明のフレーム・バッファにおけるメモリは、好ましくは、それが特にボリューム・レンダリングについて最適化されるように構成される。フレーム・バッファ２５４は、また、好ましくは、３Ｄグラフィックス・パイプライン２４２からアクセスでき、ボリュームとのポリゴン・データ２５６の混合を可能にする。

【０２３３】

図３２を続いて参照して、二重用途フレーム・バッファ２５４は、好ましくは、２つのパイプライン２４２、２５２を接続する。好ましい方法において、不透明、半透明ポリゴンの両方およびボリューム・データを持つシーンをレンダリングするために、ジオメトリ・パイプライン２４２は、まず、Ｚデプスですべての不透明なポリゴンをレンダリングする。ボリューム・メモリ２５８に記憶されたボリューム・スライスおよび半透明ポリゴンの薄いスラブを、次に、交互に（たとえば、ドプテイル式に）レンダリングする。すなわち、ボリューム・スライスをＣｕｂｅ－５パイプライン２５２で、半透明ポリゴンをグラフィックス・パイプライン２４２でレンダリングする（不透明ポリゴンも同じドプテイリング・アルゴリズムでレンダリングできるが、グラフィック・パイプラインに対する需要は増大する）。

【０２３４】

Ｚデプス・チェックは、好ましくは、隠れたオブジェクトを正しく除去するのに利用され、ブレンディングが両パイプラインにセットされ、サンプルおよび断片を正しく合成する。ジオメトリ・エンジン２４２は、好ましくは、本発明のＣｕｂｅ－５システムの要求する最終的なベースプレーン・ワープを実施する。

【０２３５】

ＤＲＡＭバッファ２５４の設計は、たとえば、256^3個のボリューム・データセットの３０Ｈｚレンダリングのために必要な毎秒５０３，０００，０００個のサンプルを得るのに重要である。したがって、まず、グラフィックス・パイプラインにレンダリング・パイプラインを接続することを説明する前に、Ｃｕｂｅ－５レンダリング・パイプラインそれ自体に対してＤＲＡＭバッファを創り出すと役に立つ。本発明のボリューム・レンダリング・システムは、好ましくは、複数のＣｕｂｅ－５パイプラインからなる。各レンダリング・パイプラインにおいて、クロック・サイクル毎に、コクセル（ＲＧＢαからなるコンポスティング・バッファ要素）を、ＳＤＲＡＭ合成バッファＦＩＦＯから読み込み、適切なコンポスティング方程式に従ってブレンドされる。次に、新しいコクセルは、ＦＩＦＯのリアに置かれる。好ましい実施例においては、コクセルの構造は、３２ビッ

トのカラー、８ビットの各ＲＧＢα、３２ビットのＺデプス情報、２４＋８ビットのステンシルを含むように変更される。この構成は、コンポスティング・ステージにおいてＺデプス・チェックを取り扱うのに必要である。任意のボリューム・レンダリングが開始する前に、不透明ポリゴン・レンダリングが完了したと仮定すると、３２ビットのＺデプス／ステンシル情報を読み込むが、書き直しはしない。したがって、あらゆるクロック・サイクルについて、各Ｃｕｂｅ－５パイプラインは、８バイトのコクセル・データを読み込み、４バイトを書き戻す必要がある。

【０２３６】

好ましくは、本発明のレンダリング・パイプラインは、１６ビットのワード・サイズを持つメモリ・チップを利用する。この構成を使用することで、４つのワードが、サイクル毎に各パイプラインによって読み出されなければならず、２つのワードを書き込まなければならない。これを行うためには、１パイプライン当たり６つの１６ビット・メモリ・インタフェースを必要とする。同期ＤＲＡＭ（ＳＤＲＡＭ）チップにおける新生の技術（本発明はそれ自体を利用することができる）が、二重データ率（ＤＤＲ）として知られており、これは、クロックの立ち上がり、立ち下がりの両方でデータを読み書きする。DDR ＳＤＲＡＭを使用することで、本発明は、１クロック当たり６４ビットのデータを読み込むための２つの１６ビット・メモリ・インタフェースを、１クロック当たり３２ビットを書き込むために、１パイプライン当たり合計３つの１６ビット・メモリ・インタフェースのために１つの１６ビット・メモリ・インタフェースを利用できる。

【０２３７】

次に図３３を参照して、クロック・サイクル毎に読み書きが実施されてパイプラインをフルに保持するので、本発明は、好ましくは、１セットのフレーム・バッファ・チップの１セット（たとえば、セットＡ）から読み出し、別のセット（たとえば、セットＢ）２６２に書き込む。Ｃｕｂｅ－５システムは、ボリュームの完全スライスについてセットA２６０から読み出し、セットＢ２６２に書き込み、次のスライスについてスワッピングすることをいとしている。しかしながらこの方法では、各フレーム・バッファ・チップ・セットは、完全なフレーム・バ

ッファを保持するのに充分に大きくなければならない。さらに、ポリゴン・エンジンは、どのセットが現セットであるかについて指示されなければならない。したがって、好ましい実施例においては、本発明は、１つのフレーム内でセットＡ２６０とセットＢ２６２の間で読み書きを交互に行い、読み込みセットが書き込みセットとなるまで読み込みセットからの処理済みコクセルをバッファする。することを交替させる２６０、そして、リードからの被処理コクセルがそれまでセットしたフレームおよびバッファ内にセットＢ２６２はライトセットになる。あらゆるメモリ・アクセスはバーストでなければならないので、各バーストは、実際には、４つのクロック・サイクルを持続させ、１６ビットDDR ＤＲＡＭチップで４つのコクセル（すなわち８個のワード）を読み書きする。Ｃｕｂｅ－５システムは、好ましくは、全部で４つのバンクを通して循環し、新しいＲＢＧα値を書き込む前にメモリ・バンド幅を飽和状態に保持する。この理由のために、好ましくは、１６コクセルＦＩＦＯ待ち行列２６４（４つのバンクの各々について４つのコクセル）に、コクセルの新しく合成されたＲＢＧα部分が記憶される。

【０２３８】

本発明のＣｕｂｅ－５ボリューム・レンダリング・システムにおけるパイプライン等の数については多くの異なった構成の可能性がある。全部で１２のメモリ・インタフェースを創り出している４つの平行パイプラインの例を図３３を参照しながら以下に説明する。図３３に示すように、各パイプラインは、フレーム・バッファのＺデプス／ステンシル部分２６８に対する１つの読み出しインタフェース２６６と、フレーム・バッファのＲＧＢα部分のセットＡ２６０、セットＢ２６２それぞれをセットするための２つの読み込み／書き込みインタフェース２７０、２７２とを含む。３０Ｈｚで256^3ボリュームをレンダリングするために、４つのパイプラインは、毎秒１２５，０００，０００個のボクセルを処理する。したがって、１３３ＭＨｚクロックが、チップおよびＳＤＲＡＭについて利用される。メモリ・チップ上へフレーム・バッファ・ピクセルをマッピングすることは、性能にとって重要である。それは、Ｃｕｂｅ－５パイプラインの処理オーダーおよび実質的に同時の４つのパイプラインによる並列アクセスに正確に整合していなければならない。説明をより容易にするために、ここで、Ｃｕｂｅ－５ア

ーキテクチャのスキュード・メモリ・アクセスが「非スキュー」にされ、ボリューム・サンプルが４つのグループにおける各スキャンラインを横切って左から右のオーダーにあるようにすると仮定する。この設計はスキュード・メモリに拡張できるが、ジオメトリ・パイプラインおよびスクリーン・リフレッシュ・システムは付加的なスキューイングに気づかなければならない。

【０２３９】

図３４は、フレーム・バッファのコクセルのＲＧＢα部分の好ましいレイアウトを示している。所与のスキャンライン２７４について、セットＢ２７８にあるピクセル・グループの前にセットAに存在するピクセル・グループがある。これはスキャンライン２７４全体を横切って繰り返される。各セットの長さは、各セットが各チップ内部の４つの異なったバンクから読み出されるピクセルを含まなければならないという事実により、６４ピクセル分である。そして、各バンクは、４つの平行チップ／パイプラインから読み出された４つのＲＧＢα値からなる。したがって、フレーム・バッファ内のピクセル・データは、８つのチップを横切ってインタリーブされる。より詳しく言えば、それは、ほんの４つのチップを横切って実際にインタリーブされる。このことは、毎秒、データの

４パイプライン×（１ＲＧＢαチップ+1デプス・チップ）×１６ビット

×１３３MHz×２データ・レート＝３４Gbits=4.15Gbytes

を読み出すインタフェースを提供する。これは、必要な

256^3×３０Ｈｚ×８ｂｙｔｅｓ＝毎秒３．７５Ｇｂｙｔｅｓ

にまさる。ここで、８バイトは、４バイトのＲＧＢα＋４バイトＺデプス／ステンシルとして構成される。さらに、フレーム・バッファ・サブシステムは、

４パイプライン×１ＲＧＢαチップ×１６ビット×１３３MHz

×２データ率＝１７Ｇｂｉｔｓ＝２．１Ｇｂｙｔｅｓ

を書くことができる。

【０２４０】

ここで再び、

256^3×３０Ｈｚ×４ｂｙｔｅｓ＝毎秒１．８Ｇｂｙｔｅｓ

を取り扱う。これは256^3ボリュームのリアルタイム３０Ｈｚレンダリングに

とって必要である。

【０２４１】

この特別なバンド幅は、シッティング・アイドルではない。スクリーンは、フレーム・バッファにおけるデータからリフレッシュされなければならない。ここで、６０Ｈｚリフレッシュ率での１２８０×１０２４スクリーン解像度を仮定し、全部で４つＲＧＢαバイトがフレーム・バッファから読み出される（本発明のバースト・モード・アクセスがどのようにでもそれらを検索するので）と仮定すると、

１２８０×１０２４×６０Ｈｚ×４ｂｙｔｅｓ＝３００Ｍｂｙｔｅｓ

が毎秒フレーム・バッファから読み出される。フレーム・バッファのＲＧＢα部分だけがリフレッシュについて必要である。したがって、リフレッシュ・データは、８つのチップから読み出される。各チップについて１０回のデータ・バースト読み出し／書き込み（セットＡかセットＢかに依存する）を行い、その後、リフレッシュのために１回データの読み出しを行えば充分である。メモリ・アクセスのこの分布により、データの一貫したストリーム（burstyではあるが）をリフレッシュ・ハードウェアに設けることができる。Ｃｕｂｅ－５パイプラインは、また、小さいパーセンテージの余分なサイクルを含み、したがって、メモリ・サブシステムが一時的にリフレッシュのために停止させられたときに、３０Ｈｚ２５６[3]レンダリングを達成する能力を失うことはない。

【０２４２】

以下、本発明の好ましい実施例に従って、Ｃｕｂｅ－５ボリューム・レンダリング・パイプラインにグラフィックス・パイプラインを接続する代替方法を説明する。この好ましいコネクション方法は、グラフィックス・パイプラインおよびボリューム・レンダリング・パイプラインの両方を常時作動状態に維持し、Ｃｕｂｅ－５チップ内のＳＤＲＡＭ合成バッファのデータを合併する。任意所与の時点で、ボリューム・レンダリング・パイプラインは、コンポスティング・バッファを通して先の薄いポリゴン・データ・スラブで現ボリューム・スライスをコンポスティングし、グラフィックス・パイプラインが、次の半透明ポリゴンの薄いスラブをレンダリングする。

【０２４３】

ここに記載される方法は、なお、上記のように、ボリューム・スライスおよび半透明ポリゴン・データの薄いスラブをドブテイルするユニークな方法を利用する。第１段階において、すべての不透明ポリゴンをベースプレーン（たとえば、スクリーンに対して最も平行なボリューム面）と一致するＺバッファ上へ投影する。次に、投影したＲＧＢαＺ画像をボリューム・レンダリング・パイプラインのコンポスティング・バッファにロードする。その後、ボリュームをコンポスティング・ステージにおいて可能にされたＺ比較でレンダリングする。半透明ポリゴンの薄いスラブは、好ましくは、ジオメトリ・パイプラインによってレンダリングされ、そして、それらの対応するＲＧＢαデータは、本発明のボリューム・パイプラインに送られ、ボリューム・パイプライン内でＳＤＲＡＭコンポスティング・バッファにブレンドされる。

【０２４４】

好ましくは、ボリューム・レンダリング・アクセラレータのコンポスティング・ステージは、１段階当たり２つの層（１つのボリュームと１つの半透明ポリゴン）を合成し、したがって、ボリューム・レンダリング・プロセスを遅延させないように改造される。これは、或る種の余分なロジックの追加を必要とする。旧コクセルｃを通して半透明ピクセル断片ｐ上にボリューム・サンプルｖを二重合成するための直接的な方程式は、５つのステージにおいて４つの加算と４つの乗算を必要としよう。

【０２４５】

【数４７】

$$C_c = C_v\alpha_v + [C_p\alpha_p + C_c(1-\alpha_p)](1-\alpha_v)$$

しかしながら、単純な数学を使用することによって、二重合成を、以下の式（いくつかの計算は再使用する）で、６つのステージにおいて４つの加算と２つの乗算を用いて計算することができる。

【０２４６】

【数４８】

$$C_o = (C_o + (C_p - C_o)\alpha_p) + [C_c - (C_o + (C_p - C_o)\alpha_p)]\alpha_c$$

当業者には明らかなように、ハードウェア・デザイナは、所与の手段にとって望ましいオプション（すなわち、より少ないロジックとより多いステージ、あるいは、より少ないステージとより多いロジック）を選ぶことになる。

【０２４７】

256^3のボリュームについてのデータ転送量を考える。好ましくは、ボリュームのフロントに２５５個のスラブ＋１つのバッファがあり、ボリュームの背後に１つのバッファがある。これら２５７個のスラブは、２５６KBのデータ（ＲＧＢαの256^2ピクセル）を含む。これは、フレーム・バッファから読み込まれ、フレーム毎に２つのサブシステム間で転送されている６４MBと同等である。３０Ｈｚフレーム率を達成することは、毎秒１．９ＧＢのバンド幅を必要とする。この多くのデータが充分に広いチャネルで転送され得るが、それは、また、フレーム・バッファから読み出されなければならない。現ＤＲＡＭフレーム・バッファの構成を変えることのなくこの多いデータを読み出すことは実質的に不可能である。さらに、フレーム・バッファは、１フレーム当たり２５７回をクリアしなければならない。

【０２４８】

このバンド幅問題を解決するために、本発明は、好ましくは、ブランク・ピクセルの実行長コード化（ＲＬＥ）を使用する。この方法では、各スキャンラインは、個別にコード化される。そして、「run-of-zeros」が、実行長に続く４つのゼロ（ＲＧＢα）としてコード化される。代表的には、シーンにおけるほんの小さいパーセンテージのポリゴンだけが半透明であるから、半透明ポリゴン・スラブは比較的まばらである。これらの薄いスラブにおけるブランク・ピクセルだけ実行長コード化することで、必要なバンド幅において９９％以上の低減を行える。好ましくは、本発明の方法は、バンド幅に節減するためにまばらな半透明ポリゴンの２Ｄ画像上のＲＬＥを利用する。

【０２４９】

この好ましい方法を使用するで、本発明のＣｕｂｅ－５システムにハードウェアを加えることが必要となる。具体的には、付加的なハードウェアは、ＲＬＥ入力ストリームを復号し、ＲＧＢα断片を創り出すことができるボリューム・レンダリング・パイプラインに含まれ得る。しかしながら、これらの断片が規則的なオーダーでボリューム・パイプラインによって利用されるので、２つのパイプラインを同期化するのに二重バッファを使用して入力ストリームを復号すると好ましい。クロック・サイクル毎に、値は、復号ハードウェアから出力される。ボリューム・レンダリング・マシンが複数のパイプラインを有する場合（現設計の大部分がこれである）、復号ハードウェアは、好ましくは、ピクセル需要を維持することができるように、各パイプラインについて複製される。

【０２５０】

同様に、ジオメトリ・パイプラインに接続される発祥端部でのＲＬＥハードウェアは、リアルタイムでデータをコード化してから、ボリューム・パイプラインに送る。しかしながら、すべての半透明ポリゴンの薄いスラブを読み出し、１フレーム当たり２５７回フレーム・バッファをクリアするのに、フレーム・バッファへの毎秒１．９ＧＢのアクセスは、なお、必要である。したがって、別々のフレーム・バッファを使用して、データをＲＬＥフォーマットで直接記憶すると好ましい。半透明データの薄いスラブが非常にまばらであるので、ラスタライジングで費やされるよりも多くの時間が、クリアリング、読み出しで費やされる。一般的にラスタライゼーションにとって最適ではないが、ＲＬＥバッファは、データのクリアリング、読み出しの両方にとって最適である。たとえば、ＲＬＥフレーム・バッファをクリアするには、256^2 フレーム・バッファ全体に書き込みする代わりに、スキャンライン毎に単一のrun of zeros（５バイト）を記憶するだけでよい。

【０２５１】

現ジオメトリ・パイプラインへのインパクトを最小限にするために、ＲＬＥフレーム・バッファは、好ましくは、ＤＲＡＭを埋め込み、それをノーマル・フレーム・バッファに並列で接続する新生技術を用いて実装される。これは、普通、

データが物理的なオーダーで与えられると仮定する普通のコード化・アルゴリズムとは異なる。しかしながら、三角形ラスタライゼーションは、断片の任意のオーダーリングを保証しない。したがって、本発明の装置は、ＲＧＢα値をデータのＲＬＥスキャンラインにランダムに挿入できなければならない。

【０２５２】

図３５は、本発明に従って形成されたＲＬＥインサートのダイアグラムを示している。各断片について、コード化されたスキャンラインは、バッファ毎にコピーされ、新しいＲＧＢα値を挿入する。ロック・サイクル毎に、単一のフリット（すなわち、ＲＧＢαピクセルまたはrun-of-zeros）が処理される。スキャンライン全体が、フリット毎に処理されると好ましい。図３５を参照して、入力バッファ（「in Buffer」）２８０が、現コード化スキャンラインを保持し、出力バッファ（「out Buffer」）２８２が、挿入した新しいＲＧＢα断片を持つ新しくコード化したスキャンラインを保持する。各サイクルで挿入すべきものの選択は、好ましくは、５バイトのマルチプレクサ２８４によって実施される。本発明の装置は、好ましくは、ポインタ、すなわち、「inPtr」２８６および「outPtr」２８８を包含する。これらのポインタは、それぞれ、in Buffer２８０、out Buffer２８２の両方の現フリットを指摘する。図３５の右側にあるロジックは、どのくらい処理されたか（「Total」）２９０と制御点の２つ（「ctrl_1」および「ctrl_3」）を計算する。他のｍｕｘ制御点（「crtl_2」）は、ＲＧＢα値（run-of-zerosについてのフラグ）のすべてを一緒に論理和演算することによって算出される。「xPos」は、断片のｘ位置として定義される。好ましくは、現バッファがｙ値毎にメモリ内に位置する場所についてのルックアップ表が実装される。したがって、新らしいピクセルを挿入しながらバッファを移動させることができ、表を簡単に更新することができる。この好ましい方法が、図３６のＲＬＥ_AddFragment擬似コード・ルーチンに示してある。図３６を続けて参照して、ＲＬＥ_AddPixelToScanline機能は、図３５に示す本発明のハードウェア実施例で生じる処理を示している。

【０２５３】

埋め込み式ＤＲＡＭを利用することによって、本発明は、処理がメモリ・チッ

プで生じるときに利用できる極めて高いバンド幅を利用する。この処理は、ＤＲＡＭ製造プロセスで実施されるのに充分に単純である。たとえば、１２８０×１０２４のフレーム・バッファの場合、必要な最大メモリ量は、５０Mbitである。これは、コード化ハードウェアについて３，０００，０００以上のゲートの余裕をもって、eＤＲＡＭダイスに適合する。

【０２５４】

図３７は、ポリゴン・パイプライン２４２およびボリューム・パイプライン２５２を、ＲＬＥフレーム・バッファ２９２を通していかに接続するかを示す好ましいブロック図である。ＲＬＥフレーム・バッファ２９２は、好ましくは、二重バッファされ、データの伝送中にレンダリングをすることを可能にある。補助フレーム・バッファが、好ましくは、断片を単に複製することによって既存のものと同じ場所で接続され、したがって、ジオメトリ・パイプライン２４２の残りの部分に影響を及ぼすことがない。ボリューム・パイプライン２５２は、また、好ましくは、二重バッファリングを利用して先のスラブをブレンドしながらデータの受信を可能にする。ここで、本発明のシステムを用いることで、ボリューム・レンダリングがポリゴンレンダリングと干渉しないことは明らかである。ボリューム・パイプライン２５２が常に反復可能なオーダー様式でメモリにアクセスするので、それが３０Hzボリューム・レンダリングを達成するのに充分な率でフレーム・バッファ内へのサンプル塗り潰し率を達成する。本発明のシステムは、グラフィックス・パイプライン２４２を利用してボリューム・メモリ２５８内に記憶されたボリュームをレンダリングする前に不透明ポリゴンをレンダリングする。これは、通常、先のフレームについてボリュームのレンダリングと同時に達成することができる。ポリゴン・エンジンがボリュームと混合した半透明ポリゴンをレンダリングしなければならない場合であっても、通常、ノーマル・シーンにおける少数の半透明ポリゴンによりボリュームが仕上がる前に不透明ポリゴンをレンダリングするに充分な時間がある。

【０２５５】

本発明の好ましい実施例によれば、プレフィルタリングでボリュメトリック・データセットに三角形を増分ボクセル化し、それによって、正確な多価値ボクセ

ル化を生成する方法が提供される。多価値ボクセル化により、相互混合したジオメトリ、正確なマルチ解像度表現および効率的なアンチエイリアシングで直接的にボリューム・レンダリングを行うことを可能にする。従来のボクセル化方法は、いずれも、バイナリ・ボクセル化だけを計算していたか、あるいは、多価値ボクセル化を非効率的に計算していた。本発明によれば、この方法は、好ましくは、各ボクセル値についてどのフィルタ機能を計算すべきかについて迅速に決定する増分式を生成する。この好ましい方法は、以下により詳しく説明するが、三角形バウンディング・ボックスの１ボクセル当たり８回の加算を必要とする。

【０２５６】

画像エイリアシングを避けるために、本発明は、好ましくは、スカラー値ボクセルを使用してボクセルの空間占有パーセンテージを表すプレフィルタリングを使用する。二次元ライン・アンチエイリアシング方法の拡張型が、従来公知である（Filtering Edges for Grayscale Displays, by S. Gupta and R. F. Sproull, Computer Graphics (SIGGRAPH81), vol. 15, no. 3, pp. 1-5, Aug. 1981）。中央差分勾配評価のための最適なボリューム・サンプリング・フィルタが表面に対して垂直な一次元向きボックス・フィルタであることも示されている。本発明の方法は、好ましくは、三角形からの距離の単純な一次関数であるこのフィルタを利用する。

【０２５７】

普通のグラフィック・ハードウェアは、これらの基本的なプリミティブの組み合わせとして表されるより高いオーダーのプリミティブで、ポイント、ラインおよび三角形をラスタライズするだけである。したがって、他のすべてのプリミティブが三角形によって表され得るために、三角形のみをボクセル化することが好ましい。ポリゴン・メッシュ、スプライン曲面、球、シリンダその他をボクセル化のために三角形に細分することができる。ポイントおよびラインは、三角形の特殊なケースであり、本発明のアルゴリズムによって同様にボクセル化することができる。中実オブジェクトをボクセル化するためには、オブジェクトの境界を一組の三角形としてボクセル化するのが好ましい。次に、オブジェクトの内部を、ボリュメトリック・ファイリング手順を使用して塗りつぶす。

【０２５８】

当業者にとって明らかなように、エッジ関数は、効果的な増分算数によって、エッジからの距離を維持するリニア表示である。本発明の方法は、この概念を３次元へ敷衍し、ボリュメトリック三角形の走査変換中にアンチエイリアシングを適用する。

【０２５９】

本質的に、本発明の三角形ボクセル化方法の一般的なアイデアは、ラスタ・オーダーで三角形のバウンディング・ボックスを走査することによって三角形をボクセル化することにある。バウンディング・ボックス内の各ボクセルについて、フィルタ方程式が、好ましくは、評価され、その結果がメモリに記憶される。式の値は、三角形からの距離の一次関数である。その結果は、好ましくは、ファジー代数ユニオン・オペレータ（すなわち、ｍａｘオペレータ）を使用して、記憶される。

【０２６０】

次に図３８を参照して、ここには、中実プリミティブ２９６の中心からライン２９４に沿った方向付けボックス・フィルタの密度分布が示してある。フィルタの幅は、Ｗとして定義される。ファジー・セット内の１つのボクセルの包含物は、ゼロと１の間で変化する。方向付けたボックス・フィルタの値によって決定されることを含む。プリミティブ２９６の表面２９８は、０．５の密度アイソ・サーフェイス上に位置すると仮定する。したがって、図３８におけるように中実プリミティブ２９６をボクセル化するとき、密度分布はプリミティブ内部の１からプリミティブ外部のゼロまで変化し、エッジのところで滑らかに変化する。表面プリミティブについて、たとえば、図３９に示す三角形３００について、密度は、好ましくは、表面で１であり、表面から距離Ｗのところでゼロまで線形に低下する。本発明は、中実体のボクセル化を意図しているが、表面のボクセル化を以下に説明する。

【０２６１】

図３９を続いて参照して、フィルタ幅Ｗについての最適値が、

$2\sqrt{3}$ボクセル

単位であることは決まっている（たとえば、Object Voxelization by Filtering, by M. Sramek and A. Kaufman, 1998 Volume Visualization Symposium, pp. 111-118, IEEE, Oct. 1998参照）。シェーディングのために、垂線が、０．５のアイソ・サーフェイスで中央差分勾配を計算することによって推定されると好ましい。中央差分フィルタの全幅がせいぜい

$2\sqrt{3}$

単位であるため、正しい勾配は、０．５の密度アイソ・サーフェイス上に見出される。三角形３００の厚さは、Ｔとして定義され得る。通常、Ｔは、厚い表面を望まない限り、ゼロである可能性がある。０．５の密度での閾値決定によって、６・トンネル・フリー・セットのボクセルが、Ｗ＝１のときに、生成される。この特性は、ボリュメトリック・フィリング（たとえば、中実オブジェクトを生成するオーダーで）について有用である。

【０２６２】

１つの三角形について非ゼロ値を持つすべてのボクセルは、好ましくは、バウンディング・ボックス内にあり、このバウンディング・ボックスは、緊密なバウンディング・ボックスよりも、すべての方向において大きいS=W+T/２ボクセル単位である。したがって、本方法の第１段階は、好ましくは、三角形３００についての緊密なバウンドを決定し、次いで、それをすべての方向においてＳボクセル単位だけ膨らませ、最も近いボクセルまで外側に丸める。

【０２６３】

図４０に示すように、頂点C_1、C_2、C_3によって定義される三角形を取り囲んでいる領域は、別々に処理しなければならない７つの領域（たとえば、Ｒ１～Ｒ７）に分割することができる。本発明の好ましい方法においては、各対象ボクセルは、７つの領域内の包含物についてテストされ、次いで、各領域について異なった式で濾過される。三角形の内部領域Ｒ１において、方向付けられたボックス・フィルタの値は、単に三角形のプレーンからの距離に比例している。三角形

のエッジに沿った領域Ｒ２、Ｒ３、Ｒ４において、フィルタの値は、好ましくは、三角形のエッジからの距離に比例する。三角形の角隅Ｒ５、Ｒ６、Ｒ７の領域においては、フィルタの値は、好ましくは、三角形の角隅からの距離に比例する。

【０２６４】

図４０を続けて参照して、領域Ｒ１～Ｒ７は、好ましくは、７つのプレーンからの距離によって区別される。第１のプレーンａは、好ましくは、三角形およびページから外側のノーマル・ベクトルａポイントと同一平面にある。次の３つのプレーンｂ、ｃ、ｄは、好ましくは、それぞれ、ノーマル・ベクトルｂ、ｃ、ｄを有し、それぞれ、三角形の角隅頂点C_1、C_2、C_3を通過する。最後の３つのプレーンｅ、ｆ、ｇは、好ましくは、三角形に対して垂直であり、エッジに対して平行である。すなわち、それらのそれぞれのノーマル・ベクトルｅ、ｆ、ｇは、三角形のプレーン内に位置し、内向きとなっており、３つすべてのプレーンからの正の距離が領域Ｒ１を定めるようになっている。プレーン係数のすべては、標準化され、その結果、ノーマルの長さが１となるが、ただし、ノーマル・ベクトルｂ、ｃ、ｄについてはそうではなくて、これらは、長さがそれぞれのエッジ長の逆数に等しくなるように標準化される。こうして、プレーンからの計算された距離は、エッジの有効長さに沿ってゼロから１まで変化する。

【０２６５】

任意の平らな表面について、表面からの任意のポイントの距離は、プレーン式係数を使用して計算することができる。

【０２６６】

【数４９】

$$Dist = \frac{Ax + By + Cz + D}{\sqrt{A^2 + B^2 + C^2}}$$

係数が事前標準化されているとき、これを簡略化すると、

【０２６７】

【数５０】

$$Dist = Ax + By + Cz + D$$

となる。この計算は、増分で行われ得るので、任意のベクトルに沿って進むときに、距離は定数分だけ変化する。たとえば、１つのプレーンからの距離が位置[x，y，z]でDistである場合、X方向における１単位距離進めることは、距離を次のように変える。

【０２６８】

【数５１】

$$\begin{aligned} Dist' &= A(x + 1) + By + Cz + D \\ &= Ax + By + Cz + D + A \\ &= Dist + A \end{aligned}$$

一般的に、任意のベクトルr = [r_x、r_y、r_z]に沿って進む場合、プレーンからの距離は、

【０２６９】

【数５２】

$$Dist' = Dist + r \odot [A, B, C]$$

ここで、

【０２７０】

【数５３】

$$\odot$$

はドット・プロダクトを示す。三角形のバウンディング・ボックスを走査する間、三角形のプレーンからの距離は、１ボクセル当たり一回だけの加算で増分計算され得る。プレーンから距離を計算するために本発明に従って実施されるこの方

法は、図４１に示す好ましい擬似コード・ルーチンによって示されている。

【０２７１】

Ｙ段階は、Ｘ段階よりも複雑である。それは、Y方向において１単位分進むばかりでなく、Ｘ方向において複数単位で後に進むからである。類似点として、リターンキーを一回押すと、紙の左マージンへ滑り戻り、行を進めるタイプライタの動作が考えられる。同様に、Ｚ段階は、Ｘ、Ｙ両方向において後に進むことと、Z方向において１単位分前方に進むことを結合する。この単純な前処理段階は、ボリューム全体にわたって効率的な進みを確実にする。数値的な近似問題が生じる場合には、各内側ループの開始時に距離値を記憶し、終了時にそれを復元し、それによって、内側ループ内の丸めエラーによる数値的なクリープを最小限に抑えることができる。

【０２７２】

多価値ボクセル化の場合、７つのプレーン距離が必要である。したがって、プレーン距離を計算するのに１ボクセル当たり７回の加算が必要となる。１ボクセル当たりの他の計算としては、ループ・インデックスを増分させること、適切な領域を決定するための比較、および、必要に応じて、密度を決定する計算がある。

【０２７３】

再び図４０を参照して、領域Ｒ１において、１つのボクセルの密度値は、好ましくは、プレーンａに垂直な向きとなっているボックス・フィルタで計算される。プレーンからの距離DistAを与えた場合、密度値Ｖは、次の式を用いて計算される。

【０２７４】

【数５４】

$$V = 1 - \frac{|DistA| - T/2}{W}$$

領域Ｒ２において、密度は、好ましくは、プレーンａ、ｂからの距離を使用して

計算される。

【０２７５】

【数５５】

$$V = 1 - \frac{\sqrt{DistA^2 + DistB^2} - T/2}{W}$$

同様に、領域Ｒ３は、プレーンａ、ｃを使用し、領域Ｒ４、プレーンａ、ｄを使用する。領域Ｒ５は、角隅ポイントC_1からピタゴラス距離を使用する。

【０２７６】

【数５６】

$$V = 1 - \frac{\sqrt{(C_1^x - x)^2 + (C_1^y - y)^2 + (C_1^z - z)^2} - T/2}{W}$$

同様に、領域Ｒ６、Ｒ７は、それぞれ、角隅ポイントC_2、C_3を使用する。

【０２７７】

隣接した三角形の共有エッジのところで、不連続またはクラックを回避すると好ましい。幸いにも、ボクセル化された表面のユニオンが正しく計算されたならば、方向付けられたボックス・フィルタは、任意の多面体についてのエッジの正確なフィルタリングを保証する。ユニオン演算子は、

【０２７８】

【数５７】

$$V_{A \cup B} = \max(V_A(x), V_B(x)).$$

での多価値密度値Ｖ（ｘ）にわたって定義され得る。他のブール演算子も利用できる。しかしながら、max演算子は、共有エッジでの正しい向きのボックス・フィルタ値を保存するので、好ましい。

【０２７９】

本発明の方法においてｍａｘを使用することの含意は、現ボクセル値をメモリから読み出し、次いで、おそらくは、修正し、メモリに書き戻さなければならないということである。したがって、１ボクセル当たり最大２つのメモリ・サイクルが必要である。

【０２８０】

本発明のアルゴリズムの効率は、不必要な計算量を制限することによってさらに向上させることができる。これは、バウンディング・ボックスが、しばしば、三角形によって影響を受けるよりも三角形によって影響されないボクセルをより高いパーセンテージで含むからである。エッジ長が所定定数を上回るとき、バウンディング・ボックスは、三角形を再帰的に細分することによってより緊密にされ得る。

【０２８１】

混在するポリゴンおよびボリュームを視覚化するために、ポリゴンは、好ましくは、目標ボリュームにボクセル化され、単一のパスでレンダリングされる。ポリゴンがボリュームに関して移動する場合、ボクセル化は、データを壊さないようにオリジナルのボリュームのコピーに生じなければならない。多価値ボクセル化されたポリゴン・ボクセルは、ボリューム・データから区別するためにタグを付けられてもよい。こうして、ポリゴンを着色し、他のデータから別個に陰影を付けることができる。

【０２８２】

上記の好ましい三角形ボクセル化アルゴリズムは、本発明のＣｕｂｅ－５ボリューム・レンダリング・システムの分散パイプラインにおいて能率的に実施される。このアルゴリズムは、ほんの少量のハードウェアを既存のパイプラインに加え、相互作用率で正確な多価値ボクセル化を実施する。請求項記載のＣｕｂｅ－５ボリューム・レンダリング・アルゴリズムの１つの重要な利点は、ボリューム・データを決定論的オーダーで筋道正しくアクセスするということである。この特徴は、このアルゴリズムについてバウンディング・ボックスの規則正しいスキャニングを可能にする。

【０２８３】

図４２において、ここには、本発明によるボクセル化パイプライン全体の好ましい実施例が示してある。オンザフライ・ボクセル化が重要であるならば、本発明のシステムは、好ましくは、ボリューム・レンダリングおよびボクセル化のための別々のパイプラインを含んでいてもよい。ボクセル化が別のパスで生じた場合、これらのボリューム・レンダリングおよびボクセル化パイプラインを結合し、ボクセル化パイプラインが、ボリューム・レンダリング・パイプラインからの大部分のハードウェアを再使用してもよい。各三角形についてのセットアップは、好ましくは、２Ｄラスタライゼーションのためにセットアップがホスト・システム上で実施されると同様の方法で、ホスト・システムで行う。

【０２８４】

図４２を参照して、パイプラインの第１ハードウェア・ステージ３０２において、７つのプレーンからの距離が計算されると好ましい。７つの単純な距離単位が、７つのプレーンの各々についての４つのレジスタに割り当てられる。好ましくは、１つのレジスタが、プレーンからの現距離を保持し、他の３台のレジスタが、はＸ、Ｙ、Ｚの段階についての増分を保持する。図４３は、本発明の好ましい実施例に従って形成された、７つのプレーンのうちの１つのための距離計算ユニット３１０を示している。この距離計算ユニット３１０は、パイプラインの距離計算ステージ３０２の一部として内蔵されてもよい（図４２参照）。他の６つのユニットは、本質的に同じ設計であるが、異なる値を保持することができる。ボクセル化の各クロック・サイクル中、パイプラインは、好ましくは、Ｘ、Ｙ、Ｚの方向のいずれかにおいて進み（すなわち、Ｘ段階３１２、Ｙ段階３１４またはＺ段階３１６を実施し）、それによって、運動の方向に従って現距離を更新する。ボリュームを通してルーピングするためのハードウェアは、既にボリューム・レンダリング・パイプラインに存在しており、したがって、三角形のバウンディング・ボックスを走査するためにここで再使用される。

【０２８５】

７つのプレーン距離を計算した後、結果の値は、好ましくは、パイプラインを流下する。図４２に示すように、次のパイプライン・ステージ３０４が、好まし

くは、現ボクセルがどの領域に存在するかについて決定する。領域選択ステージ３０４の好ましい実施例においては、いくつかのケースの交互禁制により、真理値表の結果を決定するのに、７つだけのコンパレータが必要である。たとえば、プレーンｂの負側（下側）から見た図４０において、プレーンｅからの距離に応じて、プレーンｆまたはｇからの距離をテストする必要はない。

【０２８６】

図４２を続けて参照して、領域を決定した後、次のパイプライン・ステージ３０６がフィルタ機能を計算する。現ボクセルが三角形のＳボクセル単位以内にある場合、好ましくは、パイプラインのフィルタ計算ステージ３０６が起動されるだけである。さもなければ、現ボクセルは、本質的に三角形によって影響を受けず、異なった領域は、単純なリニア表示から複雑なピタゴラス距離評価までの範囲で種々の計算を必要とする。ハードウェアが、理想的には、すべてのケースを等しく取り扱わなければならないので、このようなハードウェアが、限定解像度ルックアップ表（ＬＵＴ）によって平方根近似を実施できると好まれしい。しかしながら、入力、出力の範囲は小さく、したがって、必要なＬＵＴのサイズは小さくなる。さらに、本発明のＣｕｂｅ－５ハードウェアは、ボクセル化のために再使用され得るボリューム・レンダリングに利用できるいくつかのＬＵＴを有する。表示

【０２８７】

【数５８】

$$V = 1 - (\sqrt{Dist} - T/2)/W$$

を計算するのに３つの別個のユニットを設ける代わりに、１つのＬＵＴにすべての計算を任せるとより効率的である。この場合、入力は[0, 12]について定義されたDist^{2}であり、出力は、範囲[0, 1]における密度値Vである。

【０２８８】

７つの領域の交互禁制により、最も複雑なフィルタ計算だけのハードウェアを設けると充分である。図４０を参照して、最も複雑な計算は、領域Ｒ５、Ｒ６、Ｒ７の角隅距離計算であり、好ましい実施例では、この計算は、前述の平方根Ｌ

ＵＴに加えて、５つの加算器と３つの乗算器を必要とする。領域Ｒ２、Ｒ３、Ｒ４におけるライン距離計算はより単純であり、１つの加算器と２つの乗算器と平方根ＬＵＴだけが必要である。領域Ｒ１は、ＬＵＴへの必要な入力である二乗距離を得るのに一回の乗算を必要とするだけである。

【０２８９】

再び図４２を参照して、パイプラインの最終ステージ３０８は、好ましくは、現ボクセル値および計算した密度評価を使用してｍａｘ演算を計算する。本発明の好ましい実施例において、ｍａｘ演算子は、単にマルチプレクサに取り付けたコンパレータであって、２つ値のうち大きい方の値を守りに書き戻すようになっている。バウンディング・ボックス内の大部分のボクセルが三角形に影響を与えるほど接近していないので、メモリ・バンド幅は、必要なボクセルを読み込むだけなので節減されることになる。さらなるバンド幅節減は、現ボクセル値を変えるこれらのボクセルをメモリに書き戻すだけで達成できる。メモリにワードをリクエストすることとメモリからワードを受け取ることとの間に若干の待ち時間があるので、ボクセルは、好ましくは、パイプライン内にできるだけ早く取り込み、結果を、メモリが受け取りまで待たせる。最終ステージ３０８は、書き戻しメモリであり、これは、依存性の心配なしにバッファされる。

【０２９０】

横断を複雑にするスキューイングのコンテキスト外で本発明を説明してきた。しかしながら、本発明は、Ｙ段階、Ｚ段階距離更新値にスキューイングを構築することも意図している。スキューイングは、また、より多くの複雑さを本発明のＣｕｂｅ－５ハードウェアに加える。具体的には、最も左のボクセルをＹ方向に１単位移動させ、それをバウンディング・ボックスの外に置くとき、パイプラインは、実際には、Ｘ方向においてｐ－１段階を行ってボクセルをバウンディング・ボックス内に保持する。同様に、最も左のボクセルがＺ方向において１段階移動するとき、それは、また、負のＸ方向において１段階移動する。これは、前と同じ方法で取り扱われる。したがって、本発明の装置は、好ましくは、パイプラインが現に最も左のボクセルを処理しているとき、それを決定するのに１４より多いレジスタと対応するロジックを加えることによってスキューイングを実施し

するようになっている。

【０２９１】

プレフィルタリング（本発明のボクセル化方法と組み合わせて実施してもよい）を用いて、一連の異なった解像度のボリュームを最適に生成することができる。この技術は、異なったサイズの画像をレンダリングするのに有用である。ボリュームのサイズは、好ましくは、最終的な画像のサイズと一致するように選ぶ。こうして、エイリアシングが、すべての画像解像度で避けられ、画像スケールで見えないシーンの部分をレンダリングするという不必要な仕事が実施されない。

【０２９２】

プレフィルタリングは、さらに、模型運動ぼけにも使うことができる。たとえば、オブジェクトがカメラを通り過ぎるとき、シャッタが開いている間、複雑なボリュームを走査することなり、これが運動ぼけを生じさせる。運動ぼけを正確にレンダリングするためには、普通のレンダリング技術は、複数の画像をレンダリングし、それを単一の画像にブレンドする。しかしながら、この方法は、非常に遅い。プレフィルタリングによれば、本発明は、ボクセル化中に一度走査動作を実施し、規則的なボリューム・レンダリングと同じ時間で運動ぼけをレンダリングする。この方法は、特に運動が一定である場合（たとえば、同じ方向への移動および／または回転）には良く作動する。たとえば、飛行中に安定した速度で回転しているヘリコプタのブレードを考える。たとえば、３０Ｈｚのアニメーション・フレーム率の場合に５Ｈｚの率で回転しているブレードをボクセル化するためには、ブレードはフレーム毎に５／３０×（２π）の弧を描いて走査する。したがって、ブレードの外側エッジでは、密度値は非常に低くなり、ブレードは中心におけるよりも透明に見える。ここで、より小さいボリュームを走査すると、よりソリッドに見える。より低い密度値がより少ない不透明さを表し、より高い密度値がより多い不透明さを表すように、ボリューム・レンダリング転送機能をセットするとよい。

【０２９３】

複数のボリュメトリック・オブジェクトがオーバーラップするとき、ボリュームの投影画像は非常に複雑になる。たとえば、煙が雲から立ち昇るシーンを考え

られたい。明らかに、２つのボリュメトリック・オブジェクトは、最終フレームで結合された画像とは別個に、レンダリングすることはできない。したがって、本発明の１つの形態によって実施される好ましい方法では、複数のオブジェクトは、最終レンダリング・パスについて１つのオブジェクトに結合され、そこから画像を創り出す。

【０２９４】

２つまたはそれ以上のオブジェクトが同じスペースを占めるとき、各オブジェクトからのカラーは、投影サイト・レイに沿って各サンプル場所のところで一緒に変調されると好ましい。したがって、各オブジェクトを分類、陰影付け仕手から結合し、その後、カラー変調を行うことができると好ましい。あるいは、ボクセル・データが最初に結合された場合には、可能性のある組み合わせ毎に新しい転送機能が必要となろう。したがって、この後者の方法は好ましくない。

【０２９５】

本発明の１つの形態によれば、複数のオーバーラップしているボリュームを混合する好ましい方法は、各オブジェクトのスライスがインタレースされるように第１のオブジェクトのｚディメンジョンにおいて第１オブジェクト以外のすべｔを再サンプリングする。これは、すべてのオブジェクトを整合させる分類、シェーディング、変換を含む。オブジェクト変換は、並進、スケーリングを含み、これらは、好ましくは、最も近い隣接のコネクションおよび回転を用いて本発明の装置によって実施される。回転は、好ましくは、上述の本発明の回転方法を用いて実施される。

【０２９６】

互いに関して位置あるいは向きを変えることがないオブジェクトを含むシーンの場合、本発明は、ハイレベル・シーン・グラフ・コンパイルのような最適化を意図している。たとえば、静止オブジェクトは、一度結合してから、後のレンダリングのために保存されると好ましいが、非静止オブジェクトは、他のオブジェクトに関して移動する毎に再結合される。

【０２９７】

テクスチャ・マッピングが、高品質の画像効果（たとえば、表面ディテール、

均一照明、シャドウ）をシミュレーションする広範囲に使用されている技術である。一般的に言えば、テクスチャ・マッピングは、三次元（３Ｄ）表面上へ二次元（２Ｄ）画像をマッピングすることを含む。テクスチャ・マッピングが行われる間に、ジオメトリック・オブジェクトがスクリーン上へラスタライズされる。（ｘ、ｙ）ピクセル座標は、好ましくは、（ｕ、ｖ）テクスチャ座標にマッピングされ、ＲＧＢα値がカラー値として返されてスクリーン上のピクセル使用する。

【０２９８】

基本的には、テクスチャ・マッピングに関係しているプロセスは２つある。すなわち、（ｘ、ｙ）座標から（ｕ、ｖ）座標へのマッピングと、どんなＲＧＢαが所与の（ｕ、ｖ）座標と一致するかについての画像へのルックアップである。（ｘ、ｙ）座標から（ｕ、ｖ）座標へのマッピングは、好ましくは、当業者にとって明らかなように、単純なマトリックス乗算を含む。しかしながら、ＲＧＢα値を返すように（ｕ、ｖ）座標の画像へのルックアップは複雑である。テクスチャ・ルックアップについての超大規模集積回路（ＶＬＳＩ）ハードウェア要件は、普通、かなりのコストで、現今のグラフィックボードの大きい部分を要求する。これは、（ｕ、ｖ）座標が、希に、個別の画像座標（テクセルと呼ばれる）に直接マッピングするという事実により、本来のものである。したがって、隣接したＲＧＢα値は、好ましくは、線形に補間され、正確な（ｕ、ｖ）座標でＲＧＢα値を生成する。

【０２９９】

二次元（２Ｄ）補間は、一般的に、ピクセルが２つ以上のテクセルをカバーしない場合、充分である。しかしながら、マッピングが１テクセルより大きいピクセル範囲を生成する場合、アーティファクトが、２Ｄ補間方法を使用して画像内に導入される。高価なテクセル結合作業を避けるために、ミップ・マッピングと呼ばれる技術が、普通のグラフィックス・パイプラインによって利用され得る。ミップ・マッピングは、基本的には、画像の複数のディテール・レベル（ＬＯＤ）を記憶することからなる。次いで、（ｘ、ｙ）ピクセルが、（ｕ、ｖ）テクセルにマッピングされるとき、適切なＭｉｐ－Ｍａｐレベル・テクセルは、ピクセルがテクセルよりも小さくなるように選ばれる。より正確な方法は、ディテール・テクセルの高いレベル、低いレベルの両方から４つの隣り合うテクセルをルック

アップし、全部で８つのテクセルにトリリニア補間を実施してピクセルについて適切なＲＧＢα値を計算することである。

【０３００】

普通のグラフィックス・パイプラインからのテクスチャ・マッピング・ハードウェアは、ボリューム・レンダリングを加速するために用いられてきた。これは、Reality Engine Graphics, by K. Akeley, Computer Graphics (SIGGRAPH93), 27:109116, Aug. 1993およびAccelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware, by B. Cabral, N. Cam and J. Foran, Symposium on Volume Visualization, pp. 9 1-98, Oct. 1994のようなテキストの主題であった。しかしながら、この普通の方法は、コスト・パフォーマンスを達成もしなければ、本発明の種々の機能（たとえば、シェーディング）をサポートもしない。さらに、公知の従来技術方法を用いる場合、テクスチャ・マッピングは、データ複写なしにスケーリングすることができず、しばしば、三次元（３Ｄ）補間よりもむしろ二次元（２Ｄ）補間を使用し、データセットのダウンロードが遅く、および／またはリアルタイム４次元（４Ｄ）入力をサポートしていない。

【０３０１】

上記した本発明の好ましい形態によれば、Ｃｕｂｅ－５装置は、ジオメトリ入出力バス４６、４８（図４参照）を経て普通のジオメトリ・エンジンと結合される。好ましくは、本発明のレンダリング・パイプライン（単数または複数）は、テクスチャ・ルックアップ機能を実施するのに利用されると共に、ジオメトリ・エンジンが、（u、v）テクスチャ座標に（ｘ、ｙ）ピクセル座標をマッピングするのに使用される。簡単に言えば、ひとたびＣｕｂｅ－５装置と結合したならば、ジオメトリ・エンジンの責任は、本質的に、三角形をラスタライズすることであり、一方、本発明の装置は、好ましくは、テクスチャ・マッピングのための高性能補間エンジンを提供する。本発明の装置上でテクスチャ・ルックアップを実施するためには、テクセル・データは、好ましくは、Ｃｕｂｅ－５ユニット（単数または複数）内に含まれる３Ｄメモリにロードされる。図６Ａ、６Ｂは、２×２近傍についての３２ビット・テクセル・データが、どのようにして、１６ビッ

ト・ボクセルの2^3サブキューブに配置されるかの一例を示している。

【０３０２】

本発明の重要な別の利点は、画像ベースのレンダリングを強化する能力である。一般に、画像ベースのレンダリング方法は、複雑なシーンを、このシーンの有限の画像セットに基づいて任意の視点からレンダリングする。当業者には公知の類似した画像ベース・レンダリング方法が２つあり、すなわち、明視野レンダリングおよびルミグラフであり、これらの方法は、ソース画像のデプス情報を必要とすることなく４次元（４Ｄ）補間を用いる。

【０３０３】

図４４は、明視野レンダリングにおいて、シーンがｕｖ３２２およびｓｔ３２０プレーンによって模型化されることを示している。あらゆるｕｖグリッド・ポイントが、好ましくは，視点を定め、関連したｓｔ画像を有する。投影プレーン３２４のあらゆるピクセルについて、サイト・レイ３２６が、好ましくは、ｕｖプレーン３２２にキャスティングされる。ｕｖプレーンとサイト・レイとの交差を取り囲んでいるｕｖグリッド・ポイントに対応するｓｔ画像が、そのレイに寄与する。この寄与は、好ましくは、各ｓｔ画像にそのｕｖグリッド・ポイントを通してサイト・レイをキャスティングすることによって計算される。これらのレイは、ｓｔ画像ピクセル間にヒットし、したがって、各ｓｔ画像についてバイリニア補間を実施しなければならない。４つのレイ間の１回の最終バイリニア補間で、最終投影プレーン・ピクセル・カラーを生成する。したがって、投影プレーン３２４のあらゆるピクセルを得るには、ｓｔプレーン３２０における４回のバイリニア補間と、ｕｖプレーン３２２における１回のバイリニア補間とを必要とする。その結果、全部で５回のバイリニア補間が必要である。これら５回のバイリニア補間は、実質的に、１つの４Ｄ補間または１５のリニア補間に等しい。

【０３０４】

各投影プレーン・レイについてルックアップを実施することは、通常、ｓｔ画像へのランダム・アクセスを生じさせる。したがって、本発明の好ましい方法によれば、ｓｔ画像は、オブジェクト・オーダーでアクセスされる。このオブジェクト・オーダーは、Ｃｕｂｅ－５装置が一度だけ各ｓｔ画像ピクセルの読み出し

を許すので、本発明の装置と共に用いるのにより適っている。図４４を続けて参照して、ｕｖプレーンにおける各四辺形３２８（たとえば、ａｂｃｄ）について、４つのｓｔプレーンへの四辺形の投影像（たとえば、ａｂｃｄに対応するプレーン）は、好ましくは、４つのうちのどのタイル領域３３０が最終画像に寄与しているかを決定する。次に、すべてのｓｔタイル領域３３０が、好ましくは、４つの画像に組み立てられ、投影プレーン３２４上へ透視投影される。４つの投影画像の中の最終画像は、続いて、バイリニア補間によって形成される。補間重みは、好ましくは、オリジナル・レイおよびｕｖプレーン３２２の交差によって決定される。

【０３０５】

添付図面を参照しながら本発明の実施例を説明してきたが、本発明がこれらの実施例そのものに限定されるものではなく、本発明の範囲または精神から逸脱することなく種々の他の変形例、変更例が当業者によって実施可能であることは了解されたい。

【図面の簡単な説明】

【図１】　普通のボリューム視覚化システムのブロック図である。

【図２】　本発明の一実施例に従って形成したユニバーサル三次元レンダリング・システムを示す概念ブロック図である。

【図３】　本発明の好ましい実施を示す図２のＣｕｂｅ－５ユニットの簡略ブロック図である。

【図４】　本発明の一実施例に従って形成したユニバーサル三次元レンダリング・アーキテクチャの概要を表す機能的ブロック図である。

【図５】　本発明の一実施例に従って形成したユニット・レンダリング・パイプラインを示す機能的ブロック図である。

【図６Ａ】　本発明の好ましい方法に従って、３２ビットのテクセル・データをどのようにして１６ビット・ボクセルのミニブロックに２×２近辺について記憶するかを示すグラフ図である。

【図６Ｂ】　図６Ａの例についてのボクセル記憶、テクセル記憶の比較表である。

【図７】　本発明の好ましい順方向画像ワーピング方法に従ってソース、目標画像に特殊なパラレル保存スキャンラインを示している。

【図８】　本発明の好ましい順方向画像ワーピング方法に従ってパラレル保存スキャンライン方向を決定する方法を示すグラフ図である。

【図９】　本発明の好ましい順方向画像ワーピング方法に従ってスキャンライン処理を実施するためのソース画像におけるピクセル読み取りテンプレートを示す例の二次元グラフ図である。

【図１０】　本発明の好ましい順方向画像ワーピング方法に従って、スキャンライン処理を実施するためのバイリニア・サンプル補間を示す、図９の例の二次元グラフ図である。

【図１１】　本発明の好ましい方法に従って、目標ピクセル訂正を実施するためのピクセル値を得るサンプルについてのリニア補間の二次元グラフ図である。

【図１２】　本発明の好ましい順方向画像ワーピング方法に従ってアンチエイリアシングを実施するための異方性フィルタ・フットプリントの計算を説明するグラフ図である。

【図１３】　本発明の好ましい順方向画像ワーピング方法に従って、目標サンプル上へソース・ピクセルをスプラッティングことを説明するグラフ図である。

【図１４】　本発明の１つ方法に従って、二次元スライス・シャー分解によって三次元回転を実施するためのｙスライス・シャーの例を示す。

【図１５】　本発明の別の方法に従って、二次元ビーム・シャー分解によって三次元回転を実施するためのｘビーム・シャーの例を示す。

【図１６】　本発明のさらに別の方法に従って、二次元スライス・ビーム・シャー分解によって三次元回転を実施するためのｘスライスーｙビーム・シャーの例を示す。

【図１７】　本発明のまた別の方法に従って、三次元ビーム・シャー分解によって三次元回転を実施するための三次元ｘビーム・シャーの例を示す。

【図１８Ａ】　透視投影を実施する普通のアンダーサンプリング方法を示

す。

【図１８Ｂ】　透視投影を実施する普通のオーバーサンプリング方法を示す。

【図１９】　本発明の好ましい形態に従って、透視ボリュメトリック投影を実施する適応透視レイ・キャスティング方法を示す図であり、ビュー切頭体を、視点から指数的に増加する距離に基づく領域に分割することを示す図である。

【図２０Ａ】　本発明の好ましい適応透視レイ・キャスティング方法に従って、指数領域境界でのレイの分割／結合を示すグラフ図である。

【図２０Ｂ】　図２０Ａの適応透視レイ・キャスティング方法のレイ・セグメントＡ、Ｂ、Ｃについての効果的フィルタ重みを示すグラフ図である。

【図２１】　本発明の好ましい形態に従って、サイズ±２サンプルの二次元フィルについての重みの例を示す。

【図２２】　本発明の適応透視レイ・キャスティング方法の例を示すグラフ図であり、7^3ボリュームが視点からの３つのボクセル単位であることを示す図である。

【図２３】　本発明の１つ形態に従って、指数領域透視後前投影を実施する好ましい方法の擬似コード表現である。

【図２４】　２つの領域を横切っての、本発明の指数領域透視レイ・キャスティング方法の例を示す。

【図２５】　本発明の一実施例に従って、Ｓｏｂｅｌ勾配フィルタのｘ成分の3^3の対称近似を実施するための好ましい重みの例を示す。

【図２６】　本発明の一形態に従って単一画像においてジオメトリック・オブジェクトおよびボリュームを混合する方法を示すグラフ図である。

【図２７】　本発明の１つの形態に従って三角形を薄いスラブ境界にクリップする方法のグラフ図である。

【図２８】　本発明の好ましい形態に従って半透明のポリゴンをバケット・ソートする方法のグラフ図である。

【図２９】　本発明の１つの形態に従って、ポリゴン・フットプリントをプレワープすることによってシャード視線ジオメトリを創作する方法のグラフ図

である。

【図３０】 本発明の１つの形態に従って形成したＣｕｂｅ－５パイプラインのグラフ図であり、そこに内蔵されるＳＤＲＡＭ合成バッファを示す図である。

【図３１】 テクスチャ・メモリ、フレーム・バッファおよびジオメトリ・パイプライン間のインタフェースを概念的に示す、普通のグラフィックス・アクセラレータのグラフ図である。

【図３２】 ジオメトリ・パイプラインを本発明のＣｕｂｅ－５ボリューム・レンダリング・パイプラインと接続する二重用途ＤＲＡＭフレーム・バッファを使用している本発明の一実施例を示すグラフ図である。

【図３３】 本発明の１つの形態に従って、コクセルＦＩＦＯ待ち行列を含む各Ｃｕｂｅ－５パイプラインについてのメモリ・インタフェースを示すブロック図である。

【図３４】 本発明の好ましい実施例に従って形成した、８ドラム・チップ上のＲＧＢαコクセル・レイアウトのグラフ図である。

【図３５】 本発明の一形態に従って形成した、実行長コード化（ＲＬＥ）フレーム・バッファ・ハードウェアの埋め込みＤＲＡＭチップの部分ブロック図である。

【図３６】 本発明の１つの形態に従って、図３５のＲＬＥハードウェアで生じる処理を示す擬似コード表現である。

【図３７】 Ｃｕｂｅ－５パイプラインに含まれるＳＤＲＡＭ合成バッファにジオメトリ・パイプラインを接続するＲＬＥフレーム・バッファを示す本発明の好ましい実施例のグラフ図である。

【図３８】 本発明の１つの形態に従って、表面に対して垂直方向外方の、ソリッド・プリミティブの中心からのラインに沿って採用した指向性ボックス・フィルタの密度分布を示す。

【図３９】 本発明の別の形態に従って、三角形表面プリミティブに対して垂直なラインに沿って採用した指向性ボックス・フィルタの密度分布を示す。

【図４０】 本発明の好ましい実施例に従って。三角形プリミティブのた

めの７つのボクセル化領域の二次元図である。

【図４１】　本発明の１つの形態に従って、１つのプレーンからの距離を計算する方法の擬似コード表現である。

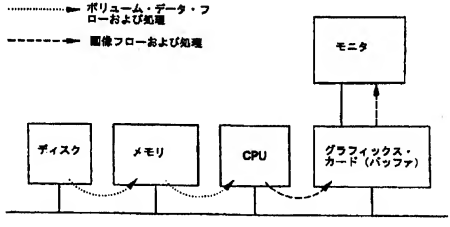
【図４２】　本発明の一実施例に従って形成した、ハードウェア・ボクセル化パイプラインの概要を示すブロック図である。

【図４３】　本発明の一形態に従って、所望のプレーンからの現ボクセルの距離を増分計算する距離ユニットを表すブロック図である。

【図４４】　本発明の１つの形態に従って画像ベースのレンダリングを実施する好ましい方法を示す頂面グラフ図である。

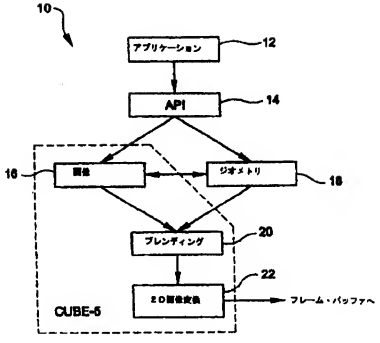
【図１】

FIG-1 従来技術



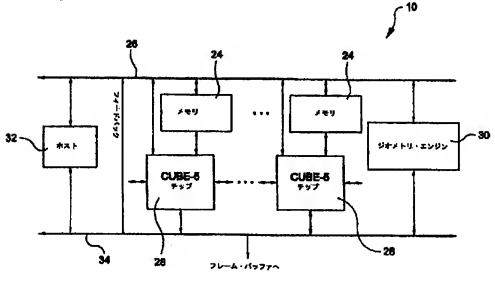
【図２】

FIG-2



【図３】

FIG-3



【図４】

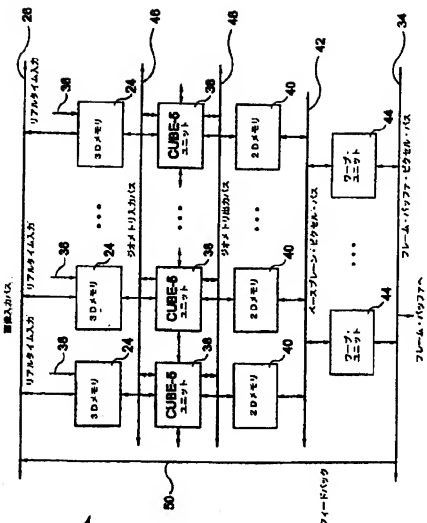
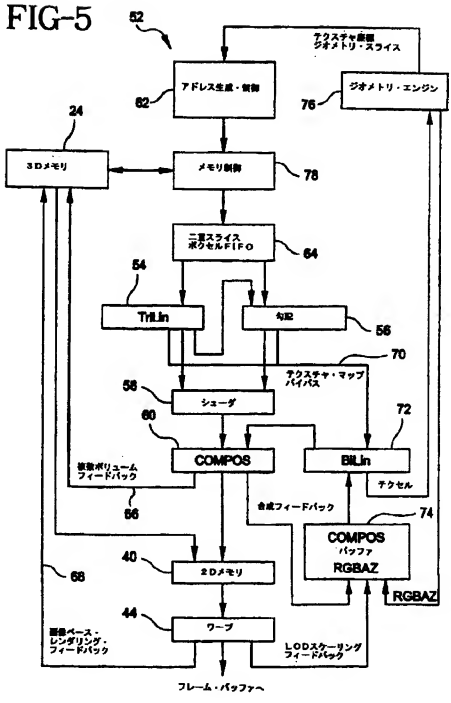

10
26
リアルタイム入力
38
3Dメモリ
24
CUBE-6
ユニット
46
ジオメトリ入力バス
ジオメトリ出力バス
40
2Dメモリ
42
ベースプレーン・ピクセル・バス
44
ワープ・ユニット
34
フレーム・バッファへ
フレーム・バッファ・ピクセル・バス
50
フィードバック

FIG-4

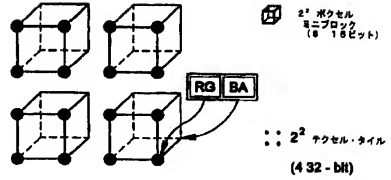
【図５】

FIG-5


52
テクスチャ座標
ジオメトリ・スライス
アドレス生成・制御
62
ジオメトリ・エンジン
76
24
3Dメモリ
メモリ制御
78
二重スライス
ピクセルFIFO
64
54
TriLin
勾配
56
70
テクスチャ・マップ
バイパス
58
シェーダ
60
72
COMPOS
BiLin
テクセル
後続ボリューム
フィードバック
66
合成フィードバック
74
COMPOS
バッファ
RGBAZ
40
2Dメモリ
68
RGBAZ
44
ワープ
画像ベース・
レンダリング・
フィードバック
LODスケーリング
フィードバック
フレーム・バッファへ

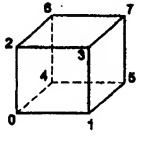
【図６Ａ】

FIG-6A



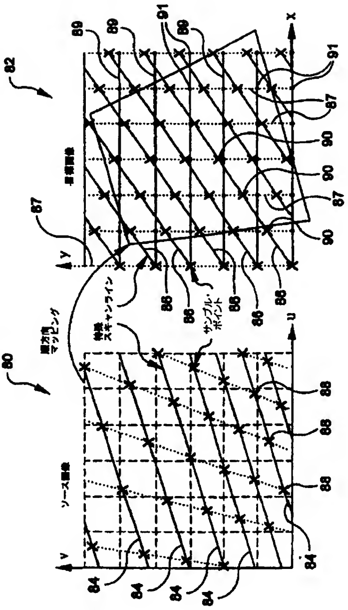
【図６Ｂ】

FIG-6B



テクセル		ピクセル
0	RG	0
	BA	4
1	RG	1
	BA	5
2	RG	2
	BA	6
3	RG	3
	BA	7

【図７】

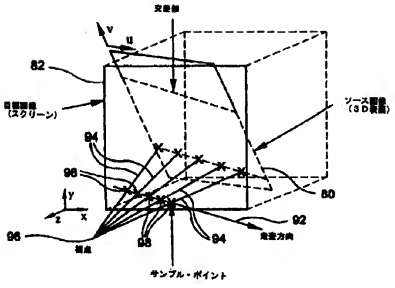


80
82
ソース画像
目標画像
順方向マッピング
特殊スキャンライン
サンプル・ポイント
84
86
87
88
89
90
91
u
v
x
y

FIG-7

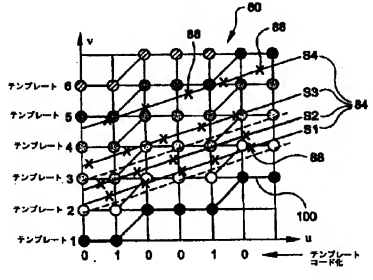
【図８】

FIG-8



【図９】

FIG-9



【図１０】

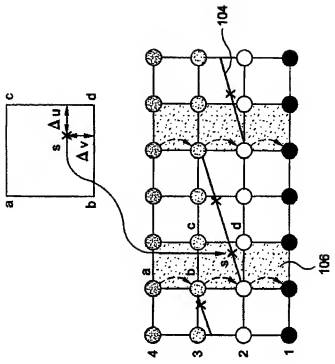
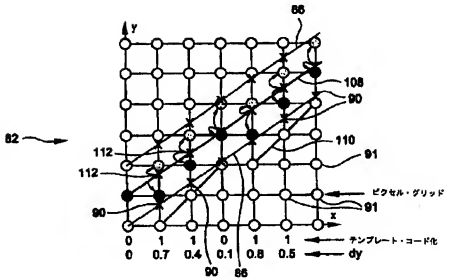


FIG-10

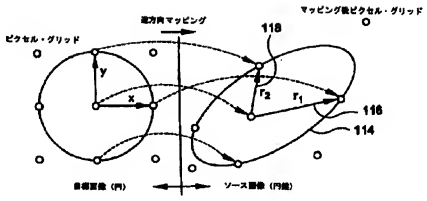
【図１１】

FIG-11



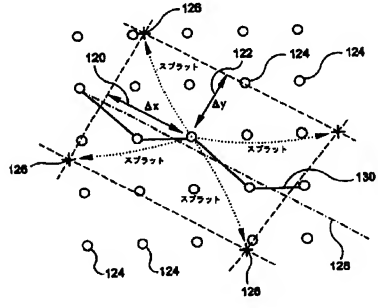
【図１２】

FIG-12



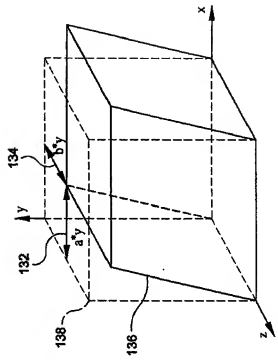
【図１３】

FIG-13



【図１４】

FIG-14



【図１５】

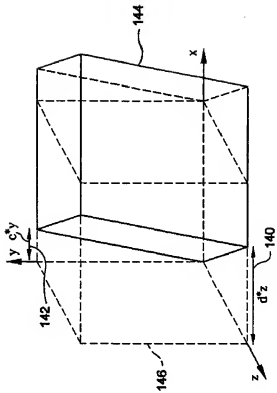


FIG-15

【図１６】

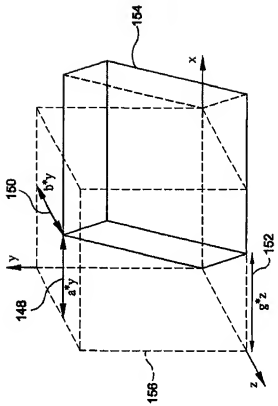
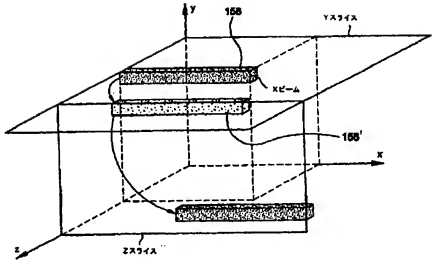


FIG-16

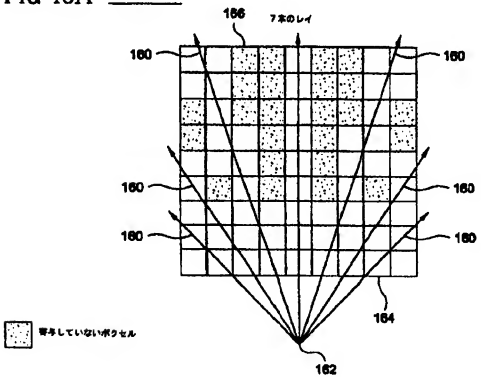
【図１７】

FIG-17



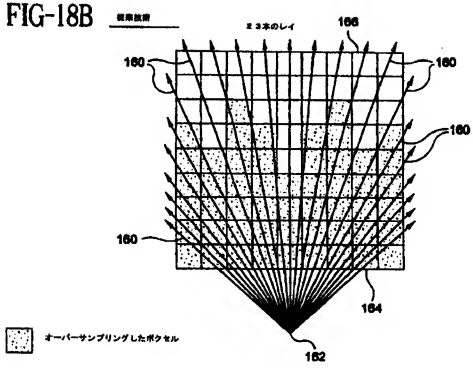
【図１８Ａ】

FIG-18A 従来技術



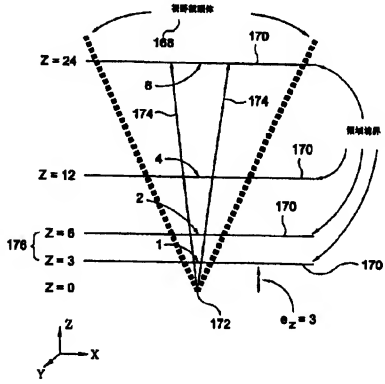
【図１８Ｂ】

FIG-18B



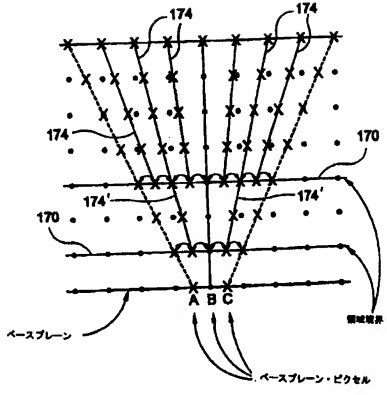
【図１９】

FIG-19



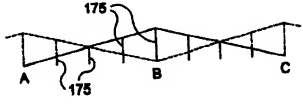
【図２０Ａ】

FIG-20A



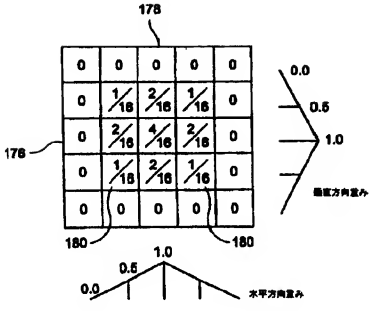
【図２０Ｂ】

FIG-20B



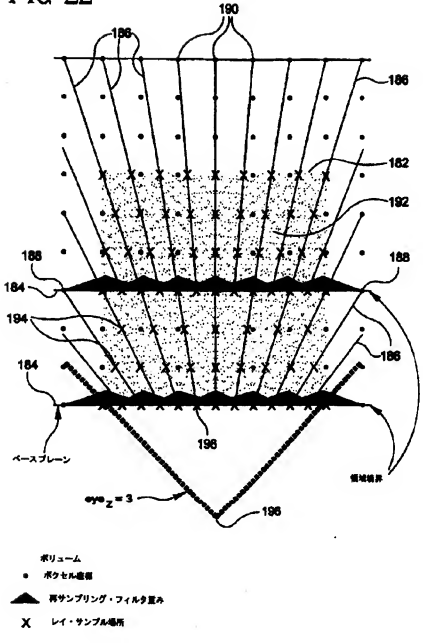
【図２１】

FIG-21



【図２２】

FIG-22


190
186
186
182
192
188
188
184
194
186
184
198
ベースプレーン
領域境界
196
ボリューム
ボクセル境界
再サンプリング・フィルタ重み
X レイ・サンプル場所

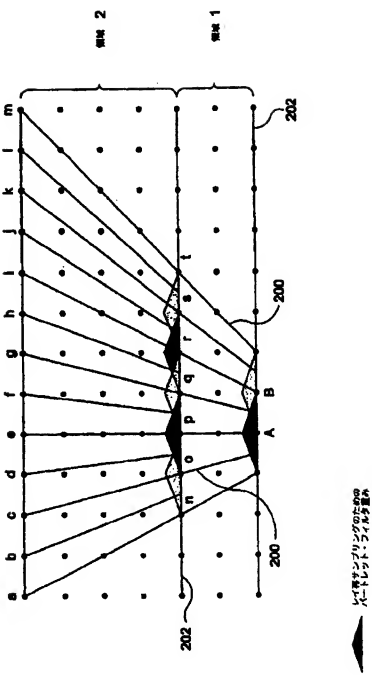
【図２３】

FIG-23

```
Compute Z-position of Eye in Voxel Units
Compute Exponential Region Boundaries
for REGION = MAXREGION to 0
    for SLICE = MAXSLICE[REGION] to MINSLICE[REGION]
        Interpolate Samples for this slice (Blinn)
        Shade and Classify Samples
        Composite Samples onto Rays in Buffer
    end for
    If not frontmost REGION
        Downsample Rays in Compositing Buffer with
            Bartlett Filter
    end if
end for
Warp Baseplane to Final Image plane
```

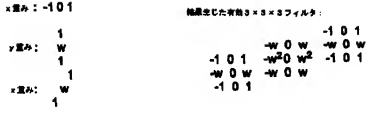
【図２４】

FIG-24


ﾚｲﾔｰ 2
ﾚｲﾔｰ 1
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
A
B
202
200
200
202
ﾚｲﾔｰｻﾝﾌﾟﾘﾝｸﾞﾊﾟﾀｰﾝ
ﾌﾟﾚｰﾄﾚｯﾄ ﾗｲﾝｸﾞﾊﾟﾀｰﾝ

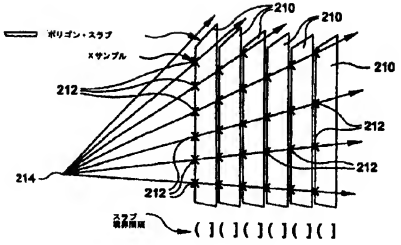
【図２５】

FIG-25



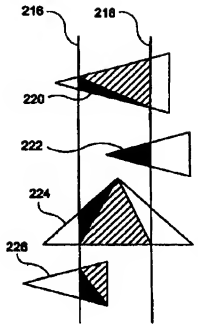
【図２６】

FIG-26



【図２７】

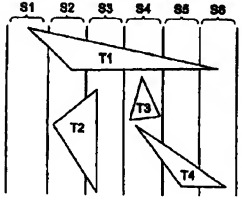
FIG-27



スライス内頂点の数	生じた三角形の数
0	2
1	1
1	3
2	2

【図２８】

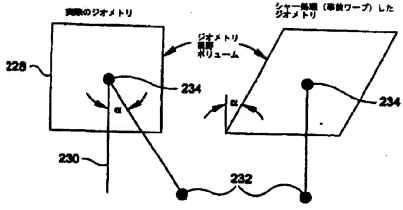
FIG-28



バケット	メンバー三角形
S1	T1
S2	T1,T2
S3	T1,T2
S4	T1,T3,T4
S5	T1,T4
S6	T1,T4

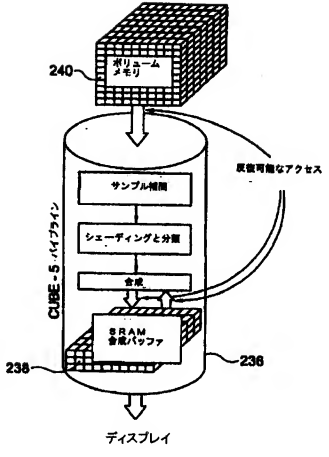
【図２９】

FIG-29



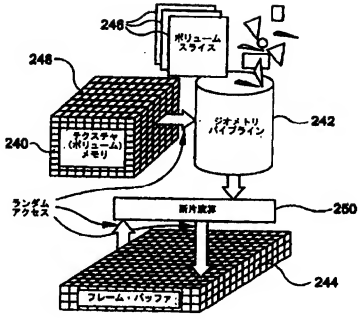
【図３０】

FIG-30


240
ボリューム
メモリ
反復可能なアクセス
サンプル補間
シェーディングと分類
合成
CUBE-5パイプライン
SRAM
合成バッファ
238
236
ディスプレイ

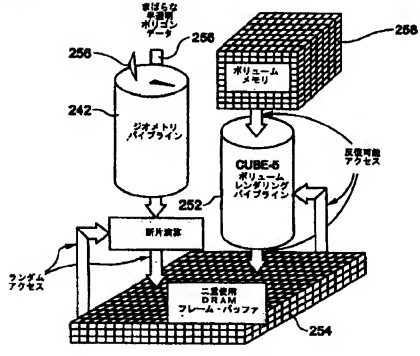
【図３１】

FIG-31 従来技術



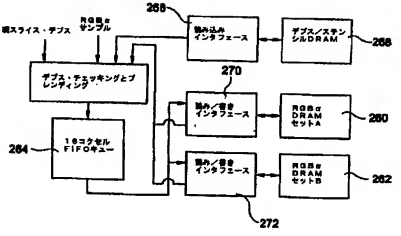
【図３２】

FIG-32



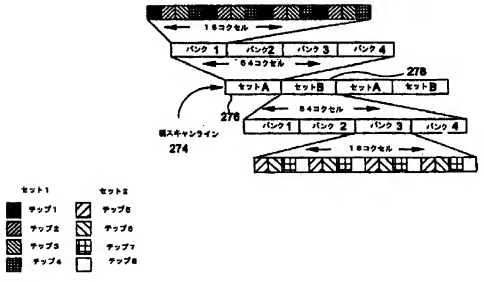
【図３３】

FIG-33



【図３４】

FIG-34



【図３５】

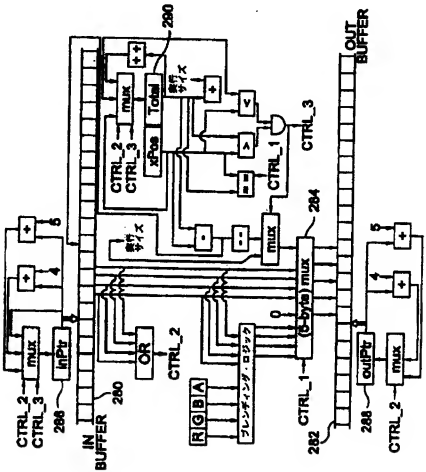

IN BUFFER
280
CTRL_2
CTRL_3
mux
inPtr
286
OR
CTRL_2
R G B A
ブレンディング ロジック
CTRL_2
CTRL_3
mux
Total
xPos
290
CTRL_1
CTRL_3
mux
284
(6-byte) mux
CTRL_1
282
OUT BUFFER
outPtr
288
CTRL_2
mux

FIG-35

【図３６】

FIG-36

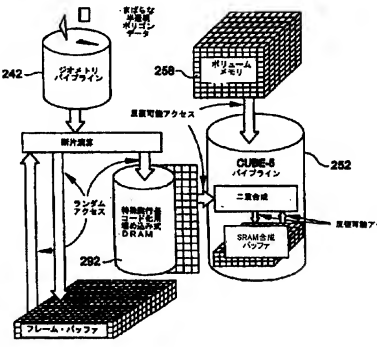
```
RLE_AddFragment (xPos, yPos, RGBA) {
    tmp = nextFreeScanline() ;
    RLE_AddPixelToScanline(data[yPos], xPos, RGBA, tmp) ;
    freeScanLine (data [yPos] ) ;
    data [yPos] = tmp ;
}

RLE_AddPixelTo Scanline(in, xPos, RGBA, out) {
    total = 0 ;
    inPtr = 0 ;
    outPtr = 0 ;
    while(total < lineWidth) {
        if (total == xPos) {
            out [outPtr: :outPtr+3] = BLEND (RGBA, in [inPtr: :inPtr+3]);
            outPtr +=4;
            total++;
            if(in[inPtr: :inPtr+3] == 0)
                in [inPtr+4] - - ;
            else
                inPtr +=4;
        }

        out [outPtr: :outPtr+3] = in [inPtr: :inPtr+3] ;
        if (in [inPtr: :inPtr+3] == 0) {
            if (total < xPos && total+in[inPtr+4] > xPos) {
                out [outPtr+4] = xPos - total -1 ;
                outPtr +=5 ;
                in [inPtr+4]  -= xPos-total ;
                total = xPos ;
            }  else {
                out [outPtr+4] = in[inPtr+4] ;
                total += in [inPtr+4] ;
                outPtr +=5 ;
                inPtr  +=5 ;
            }
        }  else {
            total ++ ;
            outPtr +=4 ;
            inPtr  +=4 ;
        } //  endif run-of-zeroes
    } //  endwhile still within scanline
}
```

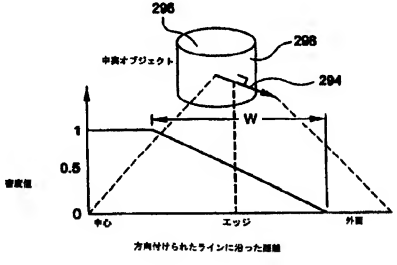
【図３７】

FIG-37



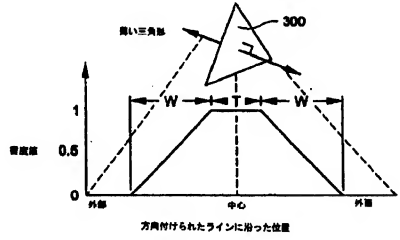
【図３８】

FIG-38



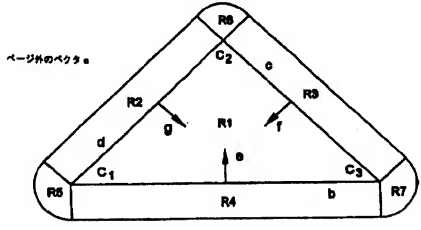
【図３９】

FIG-39



【図４０】

FIG-40



ページ外のベクタ a

【図４１】

```
Define Plane(A, B, C, D);
Find triangle bounding box(bb);
Dist = A x bb.min.x + B x bb.min.y + C x bb.min. z +
D;
xStep = A;
yStep = B - A x bb.width;
zStep = C -B x bb.height - A x bb.width;
For z = bb.min.z to bb.max.z with unit steps
    For y = bb.min.y to bb.max.y
        For x = bb.min.x to bb.max.x
            store (fDist) in [x,y,z] (voxelize)
            Dist = Dist + xStep;
        end For
        Dist = Dist + yStep;
    end For
    Dist = Dist + zStep;
end For
```

FIG-41

【図４２】

FIG-42

【図４３】

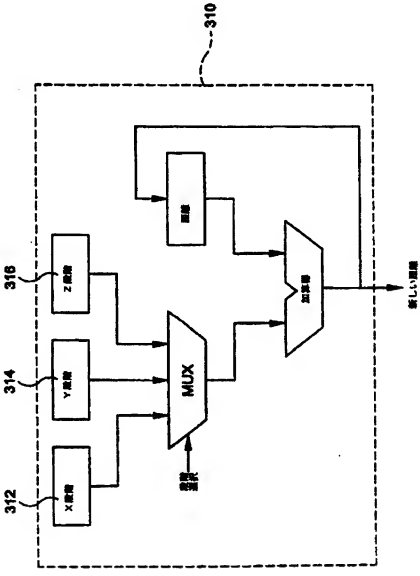


FIG-43

【図４４】

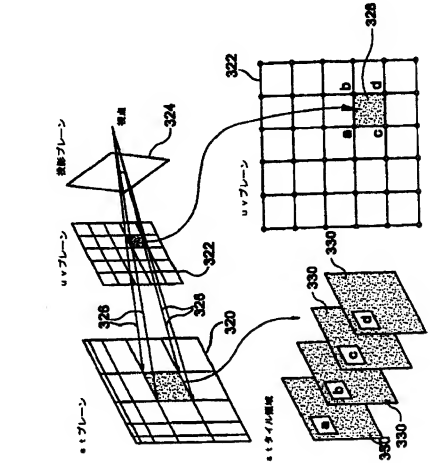

近接プレーン
視点
324
uvプレーン
322
328
326
320
stプレーン
uvプレーン
322
328
a
b
c
d
330
330
330
stタイル領域
a
b
c
d
350

FIG-44

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No. PCT/US99/16038

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) :G06T 17/00
US CL :345/419, 424
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
U.S. : 345/418, 419, 424, 420, 425

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,724,493 A (HOSOYA et al) 03 March 1998.	1-57
A	US 5,760,781 A (KAUFMAN et al) 02 June 1998.	1-57
A,P	US 5,847,711 A (KAUFMAN et al) 08 December 1998	1-57
A,P	US 5,850,226 A (NAGASAWA et al) 15 December 1998	1-57
A	US 5,877,779 A (GOLDGERG et al) 02 March 1999	1-57

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search
12 SEPTEMBER 1999

Date of mailing of the international search report
15 OCT 1999

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
CLIFFORD R. MICHAEL
Telephone No. (703) 305-9594

Form PCT/ISA/210 (second sheet)(July 1992)*

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW

(72)発明者 ビター インゲマー
アメリカ合衆国 ニューヨーク州 11790 ストーニー ブルック ストーニー ブルック ロード 1456
(72)発明者 チェン ボーカン
アメリカ合衆国 ニューヨーク州 11790 ストーニー ブルック チャピン コンプレックス 603
(72)発明者 ダシール フランク
アメリカ合衆国 ニューヨーク州 11705 ベイポート テラス ロード 546
(72)発明者 クリーガー ケヴィン
アメリカ合衆国 ニューヨーク州 11777 ポート ジェファーソン ランドール アベニュー 115

Fターム(参考) 5B057 CA01 CA08 CA12 CA16 CB01 CB08 CB13 CB16 CD20 CH05 CH14
5B080 AA17 CA04 CA05 DA07 FA03 FA08 FA17 GA04 GA22

【要約の続き】
が処理されてしまうまでレイ・キャスティング、合併／分割の段階を繰り返す段階とを包含する。本発明の装置および方法は、テクスチャ・マッピングおよび画像ベース・レンダリングを含めて、一回の画像操作で高解像度ボリューム・レンダリング、表面、ボリューム混合を行える真のリアルタイム性能を達成する。